

VŠB – TECHNICKÁ UNIVERZITA OSTRAVA

FAKULTA STROJNÍ

KATEDRA AUTOMATIZIČNÍ TECHNIKY A ŘÍZENÍ

**Diagnostika spalovacích motorů na základě měření
kolísání jejich úhlové rychlosti**

*Internal Combustion Engine Diagnostics Based on Rotational Speed
Variation Measurements*

Diplomant: Jan Benč

Vedoucí diplomové práce: Prof. Ing. Jiří Tůma, CSc.

Ostrava: 22.5.2009

Prohlášení diplomanta

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě

.....

Jan Benč

Prohlašuji, že

- byl jsem seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména §35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a §60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně ke své vnitřní potřebě diplomovou práci užít (§35 odst. 3).
- souhlasím s tím, že jeden výtisk diplomové práce bude uložen v Ústřední knihovně VŠB-TUO k prezenčnímu nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o diplomové práci, obsažené v Záznamu o závěrečné práci, umístěném v příloze mé diplomové práce, budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu §12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě

.....

Jan Benč

Jan Benč

Hlavní 194

Mikulovice

790 84

ANOTACE DIPLOMOVÉ PRÁCE

BENČ, J. *Diagnostika spalovacích motorů na základě měření kolísání jejich úhlové rychlosti* Ostrava: katedra automatizační techniky a řízení, Fakulta strojní VŠB-Technická univerzita Ostrava, 2009, 51 s. Diplomová práce, vedoucí: Prof. Ing. Jiří Tůma, CSc.

Tato práce je věnována měření kolísání úhlové rychlosti při ustálených otáčkách, nazývané též úhlové kmity za rotace a jeho praktickému využití při diagnostice spalovacích motorů. Práce se skládá ze dvou hlavních částí. První část se zabývá návrhem měřicího řetězce a jeho hardwarovým vybavením, jehož hlavními prvky jsou IRC snímač a měřicí karta NI USB-6009. Obsahem druhé části je popis softwarového zajištění měřicí úlohy. Za tímto účelem byla vytvořena aplikace zajišťující sběr a následné zpracování a zobrazení naměřených dat. V aplikaci jsou obsaženy prvky pro výpočet spektra signálu, výpočet nominální úhlové rychlosti, digitální filtry, derivace signálu a mnoho dalšího. Aplikace je vytvořena v programovacím jazyce Visual Basic a založena na platformě Microsoft .NET Framework.

ANNOTATION OF THE THESIS

BENČ, J. *Internal Combustion Engine Diagnostics Based on Rotational Speed Variation measurements*. Ostrava: Department of Control Systems and Instrumentation, Faculty of Mechanical Engineering VŠB – Technical University of Ostrava, 2009, 51 p. Thesis, head: Prof. Ing. Jiří Tůma, CSc.

Thesis deals with the measurement of the shaft angular vibration and the rotation uniformity at the quasi steady state rotation and its usage in diagnostics of engines. Thesis is break – down into two parts. The first one is dealing with the design measurement chain and its equipment with IRC sensors and the multifunction measurement card of the NI USB-6009 type. The second part is focused on the software support. For this purpose the software application was designed. The main tasks are as follows: signal acquirement, data processing and displaying output data. A user can check the validity signal using frequency spectrum. The software main functions consist of the evaluation of nominal angular velocity, digital filtration, derivation with respect to time etc. Software was created using Visual Basic, witch is based on Microsoft .NET Framework.

Obsah

Seznam použitého značení	2
1 Úvod	4
2 Návrh měřicí úlohy a její hardwarové zajištění	5
2.1 Prostředky pro měření rovnoměrnosti úhlové rychlosti	5
2.2 Měřicí karta NI USB - 6009	8
2.3 Popis měřicího řetězce	10
2.4 Princip fázové demodulace signálů	12
3 Softwarové zajištění měřené úlohy	17
3.1 Softwarová výbava NI USB - 6009	17
3.2 Sběr naměřených dat.....	21
3.3 Tachometer - nominální rychlost otáčení	24
3.4 Autospectrum – frekvenční spektrum signálu	27
3.5 Filter - digitální filtry	30
3.6 Rotation uniformity - rovnoměrnost otáčení	33
3.7 Script.....	39
3.8 Uživatelské rozhraní	42
4 Demonstrace funkčních možností	45
5 Závěr.....	48
6 Použitá literatura	50

Seznam použitého značení

$G(j\omega)$	kmitočtový přenos
j	imaginární jednotka
t	spojitý čas [s]
$x(t)$	reálná část analytického signálu
$y(t)$	imaginární část analytického signálu
$z(t)$	analytický signál
$\omega(t)$	úhlová rychlost [rad/s]
ω	reálný úhlový kmitočet [s^{-1}]
$\varepsilon(t)$	úhlové zrychlení [rad/ s^2]
$\varphi(t)$	úhel natočení [rad]
Al	Aluminium
As	Astat
CD	Compact Disc (kompaktní disk)
DFT	Diskrétní Fourierova transformace
FFT	Fast Fourier transform (rychlá Fourierova transformace)
FFTW	The Fastest Fourier Transform In The West
FIR	Finite Impulse Response (konečná impulsní odezva)
FT	Fourierova transformace
Ga	Galium
He	Helium
IC	Internal Combustions (vnitřní detonace)
IRC	Incremental rotary encoder (inkrementální snímač)
LabVIEW	Laboratory Virtual Instruments Engineering Workbench
LDVs	Laser Doppler Vibrometers (laserové dopplerovské vibrometry)
LED	Light emitting diode (elektroluminiscenční dioda)
MAX	Measurement and Automation Explorer
Ne	Neon
NI	National Instruments
OS	Operační Systém
PC	Personal Computer (osobní počítač)
PZ	Polopropustné zrcadlo

RMS	Root Mean Squared (Efektivní hodnota)
RPM	Revolution Per Minute (Otáčky za minutu)
TEDS	(Transducer Electronic Data Sheet)
TTL	(Transistor – Transistor - Logic)
USB	Universal Serial Bus (univerzální sériová sběrnice)
Z	Zrcadlo

1 Úvod

Diagnostika v oblasti spalovacích motorů prochází neustálým vývojem. Důvodem tohoto rozvoje jsou především ekonomické ukazatele. Použitím vhodného diagnostického systému vede ke zkrácení doby potřebné pro odhalení poruchy, což má za následek snížení nákladů na lidskou práci. Diagnostické systémy nám také poskytují možnost průběžného monitorování chodu motoru a případnou závadu tedy můžeme odhalit již v prvopočátcích jejího vzniku a eliminovat tak následná poškození dalších částí, nebo konstrukčních celků. Tím předejdeme nákladným a rozsáhlým opravám.

Tato práce bude zaměřena na problematiku diagnostiky spalovacích motorů na základě měření rovnoměrnosti otáčení klikového hřídele při ustálených volnoběžných otáčkách. Ustálené otáčky však nejsou ideálně stálé. Vždy dochází k malým změnám vlivem proměnlivého hnacího momentu motoru, které se projevují jako úhlové kmity za rotace. Měřením a následným zpracováním takto nerovnoměrného otáčení jsme schopni získat informace o kvalitě pochodu motoru a popřípadě detekovat poruchy.

Práce se skládá ze dvou hlavních částí. První část se zabývá návrhem měřicího řetězce a jeho hardwarovým vybavením. Zde se bude nutné seznámit s problematikou měření úhlové rychlosti pomocí IRC snímačů a Dopplerovského laseru. Popsány budou základní vlastnosti, fyzikální princip či způsob detekce směru otáčení. Další studium se bude ubírat k měřicí kartě NI USB-6009, zde budou uvedeny její základní vlastnosti spolu s popisem vstupních a výstupních kanálů. Prostudovány budou též metody pro měření rovnoměrnosti otáčení. Následovat bude návrh měřicího řetězce, včetně jeho hardwarového vybavení.

Druhá část se zabývá softwarovým zajištěním měřicí úlohy. První kroky zde povedou k studiu softwarové výbavy měřicí karty. Dále k seznámení se s vývojovými prostředími vhodnými pro tvorbu měřících aplikací, podporující zvolený typ měřicí karty. Nabyté vědomosti poté využít při tvorbě vlastní aplikace pro sběr a zpracování naměřených dat. Následovat bude demonstrace funkceschopnosti diagnostického systému, zhodnocení dosažených výsledků a návrh dalšího řešení.

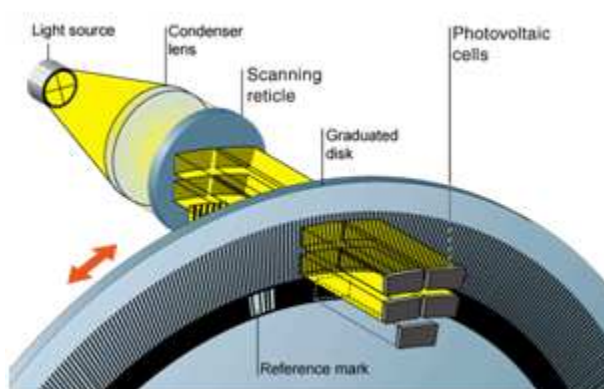
2 Návrh měřicí úlohy a její hardwarové zajištění

Základem k úspěšné realizaci měřicí úlohy je nejen volba správného postupu měření, ale i vhodná volba technických prostředků tak, aby měření bylo realizováno v požadované přesnosti s co možná nejmenšími náklady na realizaci daného problému. Návrhem technických prostředků potřebných pro realizaci naší úlohy se bude zabývat tato kapitola.

2.1 Prostředky pro měření rovnoměrnosti úhlové rychlosti

Pro měření úhlových kmitů se v minulosti používaly hlavně torziografy se setrvačnou hmotou. Ta byla svázána torzní pružinou s hřídelí, jejíž nerovnoměrné otáčky byly zkoumány. Měřením vzájemných torzních posunů mezi hřídelí a touto setrvačnou hmotou se dospělo k signálu, který charakterizoval úhlové kmity. Dalším způsobem měření úhlových kmitů jsou tangenciálně montované akcelerometry na rotoru. V současnosti se s výhodou využívají také IRC snímače a dopplerovské lasery a to zvláště pro jejich vysokou rozlišovací schopnost.

Inkrementální snímač je základním prvkem pro měření polohy, úhlu natočení, nebo úhlové rychlosti. V jednoduché, ne příliš přesné variantě, jsou snímače obsaženy např. v běžných počítačových myších. Precizní provedení těchto snímačů pak nalezneme např. na obráběcích strojích. Jejich charakteristické rysy jsou velká rozlišovací schopnost (až tisíce pulzů za otáčku), malé rozměry a malá hmotnost. Jejich konstrukce se vyznačuje zvýšenou odolností vůči vibracím a rázům, což umožňuje jejich využití při měření v náročných provozních podmínkách. Pomocí běžných rozhraní je možné přímé připojení na stávající elektronické obvody, nebo řídicí jednotky.

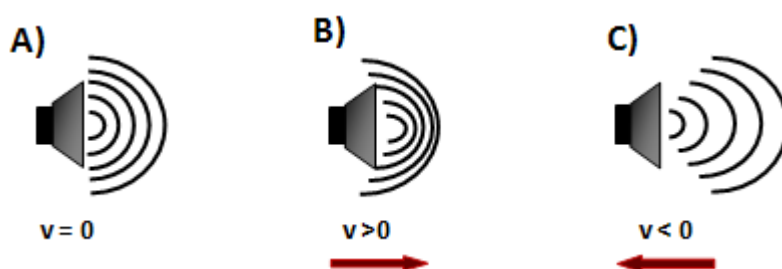


Obr. 2.1 Fotoelektrické snímání metodou prosvětlování [HEIDENHAIN 2008]

Jádro inkrementálního snímače, založeného na fotoelektrickém principu, je znázorněno na obr. 2.1. Měřicí princip těchto senzorů je založen na clonění světelného toku

mezi světelným zdrojem (LED diodami) a fotodetektozem (fototranzistor, nebo jiný fotocitlivý prvek). Princip lze demonstrovat na funkci jedné ze čtyř pevných mřížek, proti níž se otáčí mřížka umístěná na optickém disku. Nacházejí-li se obě mřížky ve shodné poloze, prochází jimi maximum světelného toku. Ten je snímán fotodetektozem, jehož výstupní signál dosáhne vlivem světelného toku maxima. Pootočí-li se pohyblivá mřížka na disku jen o nepatrný úhel, nastane mírné zaclonění a na fotodetektoru se v důsledku snížení světelného toku sníží i úroveň signálu. Při dalším otáčení bude signál na fotodetektoru nadále slábnout, až dosáhne nuly. V tom okamžiku je zcela přerušeno světelné spojení. Tím vzniká signál kvazisinusového průběhu, jehož frekvence je závislá na rychlosti otáčení. Tyto kvazisinusové průběhy jsou stejně jako mřížky vzájemně posunuty o 90° . Abychom získali signál symetricky rozložený vůči nulové ose, použijeme diferenciální metodu. Obě tyto kvazisinusové křivky jsou pomocí napěťových komparátorů proměněny na pravouhlé signály A a B, které jsou vůči sobě rovněž fázově posunuté o 90° . Poté jsou zesíleny na potřebnou úroveň a přivedeny na kanály A (A-invertovaný) a B (B-invertovaný). Dalším kanálem je 0 (0 index invertovaný), jehož výstupem je pravouhlý signál dávající informaci o počátku otáčení. V současné době je často optický princip snímání nahrazován principem magnetorezistivním. Zde je optický kotouč nahrazen mnohapólovým prstencovým permanentním magnetem, a fotodetektor Hallovými senzory. Výstupem snímače je následně dvojice (A a B) popř. trojice (A, B a Z) signálů, zcela analogická signálům klasického optického inkrementálního snímače. [HEIDENHAIN 2008, Tůma 2002]

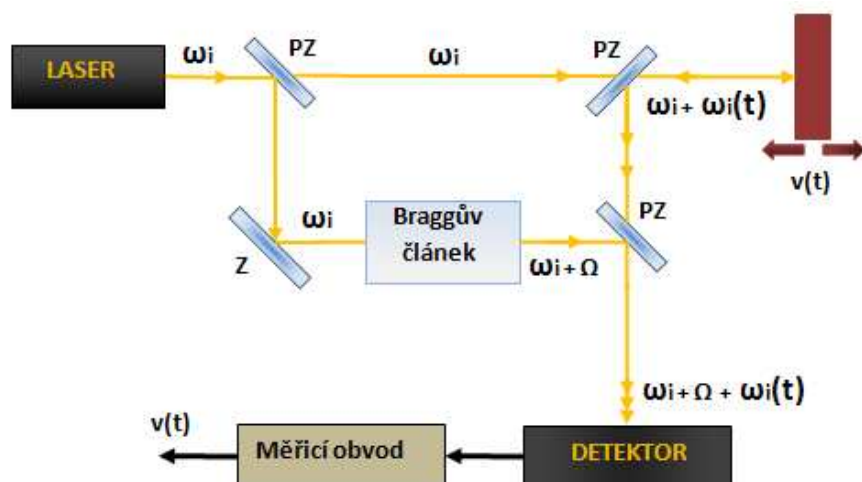
Laserové vibrometry založené na Dopplerově jevu (dále jen LDVs) se používají pro bezdotyková měření, kde standardní akcelerometry nelze z technických důvodů použít, nebo nedosahují požadovaných parametrů. LDVs jsou schopné měřit vibrace až do 30 MHz s velmi lineární fázovou odezvou a velmi malou nejistotou měření. Dopplerův jev je známější spíše z akustiky, kdy frekvence vlnění zjištěná pozorovatelem je jiná, než frekvence zdroje a to za předpokladu že vzdálenost mezi zdrojem a pozorovatelem se mění v čase.



Obr. 2.2 Dopplerův jev

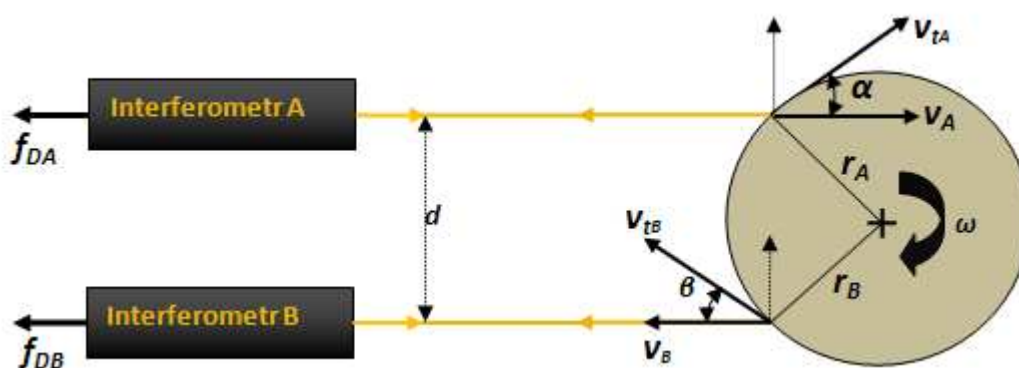
Při vzájemném přibližování je tedy přijímaná frekvence vyšší (viz obr. 2.2 příklad B) a naopak (viz obr. 2.2 příklad C). Princip laserových vibrometrů je založen na interferenci

dvou koherentních, lineárně polarizovaných světelných paprsků. Pro zdroj paprsku jsou užívány nízkovýkonové lasery na bázi Ga - Al - As, nebo He - Ne. Pomocí laserových interferometrů lze přímo měřit amplitudu kmitů čítáním referenčních proužků (např. Michelsonův interferometr). To však lze využít pouze při nízkých rychlostech, nebo při metrologii akcelerometrů. [KREIDL 2006, POLYTEC 2008, HALLIWELL 2008]



Obr. 2.3 Laserový interferometrický vibrometr

Příklad upořádání heterodynního laserového interferometru je na obr. 2.3. Referenční paprsek se na prvním polopropustném zrcadle dělí na dva. První část paprsku prochází Braggovým deflektorem, který konstantně posune jeho kmitočet o Ω . Tento konstantní posun kmitočtu poté tvoří základní kmitočet signálu při nulových otáčkách. Je tedy důležitý k rozpoznání směru vibrací. Druhá část referenčního paprsku projde přes druhé polopropustné zrcadlo a pomocí čočky je směřována na povrch měřeného objektu. Zpětně odražený paprsek, který je na základě Dopplerova jevu frekvenčně modulovaný, se na polopropustném zrcadle odráží směrem k detektoru. Zde spolu s referenčním svazkem difraktovaným Braggovým modulem interferují. Výsledným signálem po fázové demodulaci v elektronické části přístroje je modulační signál $v(t)$, který může mít zcela obecný periodický průběh.



Obr. 2.4 Aplikace laserového vibrometru na rotující hřídeli

Laserové interferometrické vibrometry jsou vhodné i pro měření rotujících hřídelů. Zde je měřena tangenciální složka vektoru rychlosti ve směru měřicího optického paprsku. Pro tyto případy se používají dva interferometry, které jsou od sebe vzdáleny o konstantní vzdálenost d . Princip činnosti je presentován na obr. 2.4. Výstupem takového uspořádání je informace o úhlové rychlosti rotujícího tělesa.

Pro výpočet Dopplerovy frekvence platí:

$$f_D = f_{DA} + f_{DB} = \frac{2d \omega}{\lambda} \quad (2.1)$$

Pro jednotlivé výstupy interferometrů poté platí:

$$f_{DB} = \frac{2 \cdot v_B}{\lambda} = \frac{2(\omega r_B \cos \beta)}{\lambda} \quad (2.2)$$

$$f_{DA} = \frac{2 \cdot v_A}{\lambda} = \frac{2(\omega r_A \cos \alpha)}{\lambda} \quad (2.3)$$

Jejich vzdálenost je určena vztahem:

$$d = r_A \cos \alpha + r_B \cos \beta \quad (2.4)$$

Dále je nutné ještě dodat, že mezi vzdáleností d a rozsahem měřicího přístroje platí nepřímá úměra. Důležitý je také sklon optické hlavy vzhledem k ose rotace měřeného objektu. Pokud spolu nesvírají pravý úhel je snižována efektivní hodnota RPM. Následným zpracováním Dopplerova frekvenčního výstupu je možno měřit úhlové či torzní kmity, nebo úhlovou rychlost a zrychlení. [KREIDL 2006, POLYTEC 2008, HALLIWELL 2008]

2.2 Měřicí karta NI USB - 6009

Logickým krokem po zvolení vhodných snímačů je volba vhodné měřicí karty. Pro naši úlohu byla zvolena měřicí karta od firmy National Instruments s označením NI USB-6009. Tato karta disponuje nejen nízkou cenou, ale i širokou škálou funkčního vybavení a podporou mnoha operačních systémů.



Obr. 2.5 Měřicí karta NI USB-6009 [KMIT, P 2008]

Karta je dodávána s podrobným návodem, jednoduchými vzorovými aplikacemi a vlastním měřicím softwarem. Velkou nevýhodou této karty je absence antialiasingového filtru. Ten bude muset být v případě potřeby zajištěn softwarově. V následující části budou shrnuty nejdůležitější vlastnosti:

- Sběrnice USB
- Podpora systémů Windows, Linux, Mac OS, Pocket PC
- Vzorkovací frekvence 48 kS/s
- Analogové vstupy 8 SE/4 DI
- Analogový vstup 14 (bitu)
- Vstupní rozsah od ± 1 do ± 20 V
- Digitální trigger

[NATIONAL INSTRUMENTS 2008]

Měřicí karta je rozdělena na dvě poloviny. Prvních šestnáct pinů je určeno pro analogové signály. Z toho osm pinů reprezentuje vstupní kanály, které při diferenciálním zapojení tvoří vstupy pro čtyři analogové signály. Toto zapojení je charakterizováno vyšší přesností a volbou měřicího rozsahu od ± 1 V až do ± 20 . Pokud potřebujeme měřit více než čtyři analogové signály, využijeme zapojení „single-ended“. Pomocí tohoto zapojení je možné měřit až osm vstupních signálů. Na každý pin tak připadá jeden vstupní kanál se společnou referencí na GND. Možnost měření více analogových signálů je zde vykoupena sníženou přesností a fixním rozsahem ± 10 V. Další dva piny reprezentují výstupní kanály. Pro generování výstupních kanálů je zde použit dvanáctibitový DA převodník s postupnou aproximací. Rozmezí výstupního signálu se pohybuje od 0 V do 5V. Pro doplnění bude ještě uvedeno, že každé analogové měření může být započato pomocí softwarového či digitálního triggeru.

[NATIONAL INSTRUMENTS 2008]

Tabulka 2.1 Přesnost diferenciálního zapojení při zvoleném rozmezí

Rozmezí	Absolutní přesnost při 25 °C [mV]	Maximální přesnost při 0 – 50 °C [mV]
± 20	14.7	138
± 10	7.73	84.8
± 5	4.28	58.4
± 4	3.59	53.1
$\pm 2,5$	2.56	45.1
± 2	2.21	42.5
$\pm 1,25$	1.70	38.9
± 1	1.53	5

Dalších šestnáct pinů je určeno pro digitální signály. Zde je mnohem větší volnost mezi počtem vstupních a výstupních signálů. Pro tyto účely je zde vyhrazeno dvanáct pinů z nich každý může tvořit vstupní či výstupní kanál. Vše závisí na volbě uživatele. Dále zde najdeme dva piny sloužící jako zdroj stejnosměrného napětí na úrovni +2.5 V a +5 V. Poslední pin Označený jako PFIO slouží jako vstup pro interní čítač nebo pro hardwarový trigger. Další údaje určující digitální vstupy a výstupy:

- Pracovní hladina od -0,3 V do 2V nebo od 0,8 V do 5,8 V
- Kompatibilní s CMOS, TTL, LVTTTL
- 12 Vstupů/výstupů
- Rozlišení čítače 32 bits
- Maximální vstupní frekvence pro čítač 5 MHz
- Minimální šířka pulzu 100 ns

[NATIONAL INSTRUMENTS 2008]

GND	1	17	P0.0
AI 0/ AI 0+	2	18	P0.1
AI 4/ AI 0-	3	19	P0.2
GND	4	20	P0.3
AI 1/ AI 1+	5	21	P0.4
AI 5/ AI 1-	6	22	P0.5
GND	7	23	P0.6
AI 2/ AI 2+	8	24	P0.7
AI 6/ AI 2-	9	25	P1.0
GND	10	26	P1.1
AI 3/ AI 3+	11	27	P1.2
AI 7/ AI 3-	12	28	P1.3
GND	13	29	PFIO
AO 0	14	30	+2,5 V
AO 1	15	31	+5 V
GND	16	32	GND

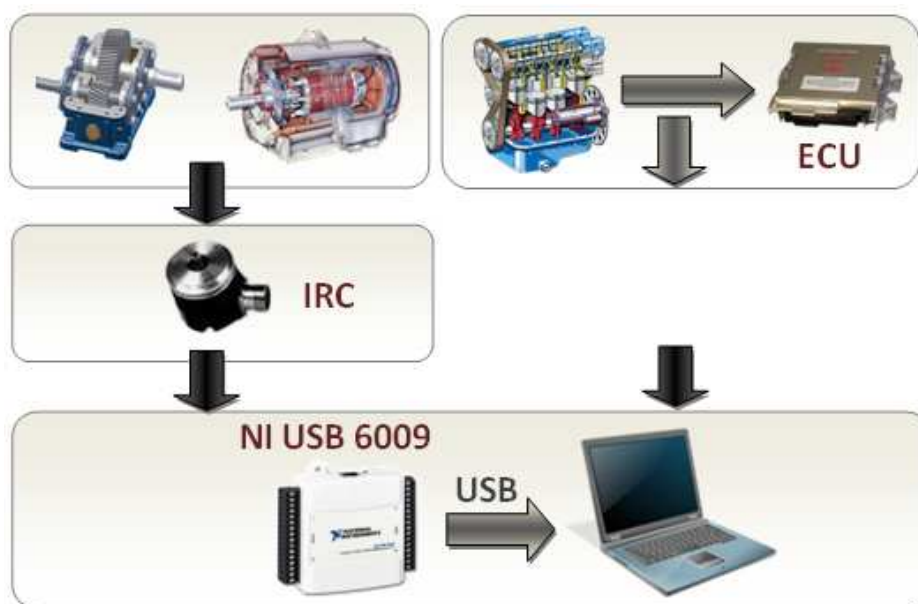
Obr. 2.6 Označení jednotlivých pinů Měřicí karty NI USB 6009

2.3 Popis měřicího řetězce

S ohledem na univerzálnost je měřicí úloha navržena obecně tak, aby ji bylo možné realizovat na jakémkoliv zařízení, které vykonává rotační pohyb při ustálených otáčkách. Další podmínkou realizace je možnost připevnění IRC snímače na námi měřenou část zařízení. Do této oblasti spadají například již dříve zmíněné spalovací motory, převodovky, elektromotory a mnoho dalšího. Za účelem měření natočení bude použit IRC snímač, jehož volba bude závislá na povaze měřené úlohy. Vhodnější jsou spíše IRC snímače s nižším

Diagnostika spalovacích motorů na základě měření kolísání jejich úhlové rychlosti

rozlišením, řádově s desítkami, až stovkami pulzů za otáčku. Volbu snímače zde ovlivňují především faktory jako rychlost otáčení měřené rotující části zařízení a vzorkovací frekvence měřicí karty. Maximální rychlost otáčení, nebo počet pulzů se dá odvodit vztahem 2.5 a 2.6. Pro vyhodnocení rovnoměrnosti otáčení bude použita metoda fázové demodulace, jejíž popis je taktéž součástí této kapitoly.



Obr. 2.7 Návrh měřicího řetězce

Vztah pro určení frekvence otáčení:

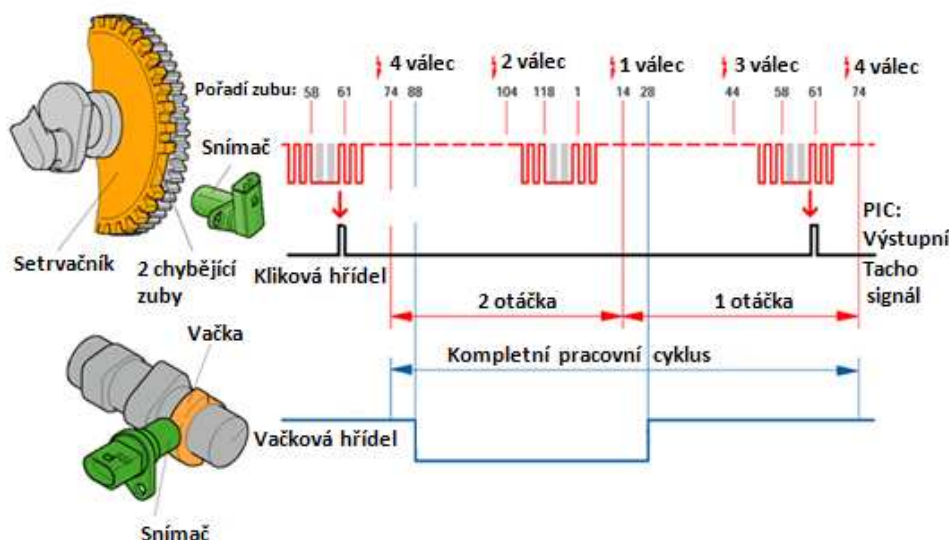
$$f_{rot} = \frac{RPM_{zař}}{60} \text{ [Hz]} \quad (2.5)$$

Zde číselník zlomku $RPM_{zař}$ vyjadřuje rychlost otáčení [ot. /min]

Vztah pro určení minimální vzorkovací frekvence:

$$f_{samp} > f_{rot} \cdot n_{IRC} \cdot 10 \text{ [Hz]} \quad (2.6)$$

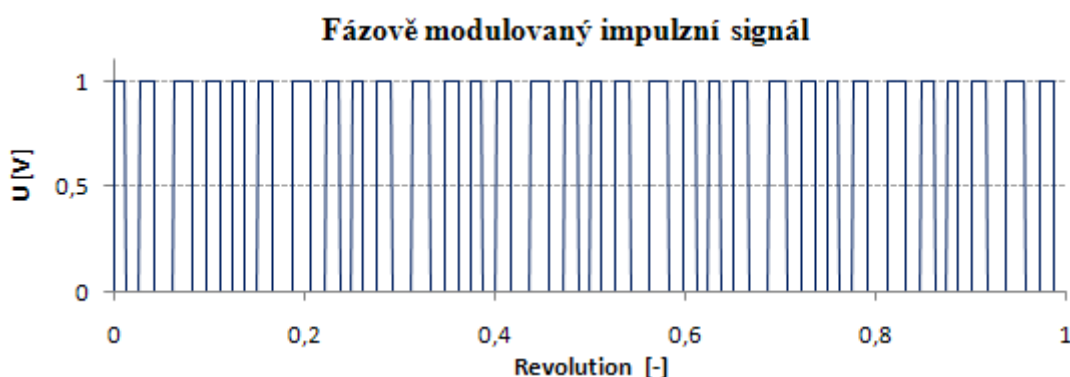
Zde parametr n_{IRC} vyjadřuje počet pulzů za otáčku a poslední koeficient byl zvolen tak, aby na jeden pulz vycházelo nejméně pět vzorků. To umožní optimální použití metody fázové demodulace. Při aplikaci měření rovnoměrnosti otáčení na spalovací motor, je velice výhodné využít již zabudovaných senzorů. Zdrojem IRC signálu zde bude snímač umístěný na setrvačnicku. Ten generuje signál pro řídicí jednotku motoru ve tvaru $60 - 2 = 58$ pulzů. Chybějící pulzy jsou dány absencí dvou zubů, v jejich lineárním rozložení. Tyto chybějící pulzy by mohli znamenat při použití metody fázové demodulace určitou chybu. Budou tedy interpolací doplněny. Jelikož se čtyřdobé spalovací motory otáčejí pouze jedním směrem, nebude potřeba vyšetření směru otáčení. Nominální úhlová rychlost zde bude získána pomocí pulzního signálu ze snímače na vačkovém hřídeli, jehož výstup je při jedné otáčce na nulovém potenciálu a následující otáčce na úrovni TTL.



Obr. 2.8 Ukázka výstupních signálů vybraných senzorů umístěných na motoru

2.4 Princip fázové demodulace signálů

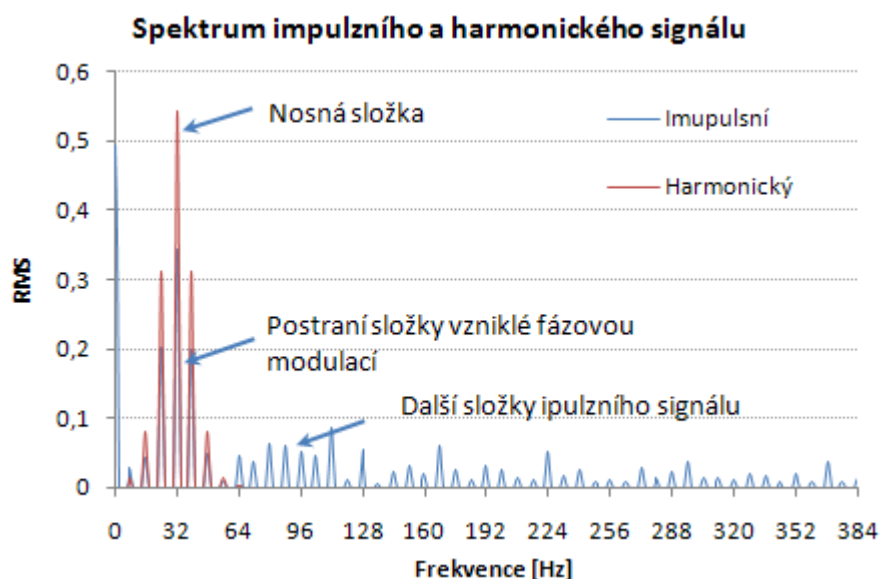
Při otáčení obdržíme na snímači impulzní signál s tvarem blízkým obdélníku. Ideální příklad tohoto signálu je presentován na obr. 2.9. V případě rovnoměrného pohybu (rychlost otáčení je konstantní) tento signál obsahuje harmonické složky příslušející složce o základní frekvenci impulsu. Tyto složky se nazývají nosnými. Při jakékoliv nepatrné změně rychlosti poté dochází k fázové modulaci tohoto signálu, která se projeví vznikem postranních složek k složkám nosným. Jako nosnou složku je dobré vybrat složku se základní frekvencí impulsů. Ta je v daném spektru nejvyšší. [Tůma 2002]



Obr. 2.9 Fázově modulovaný impulzní signál

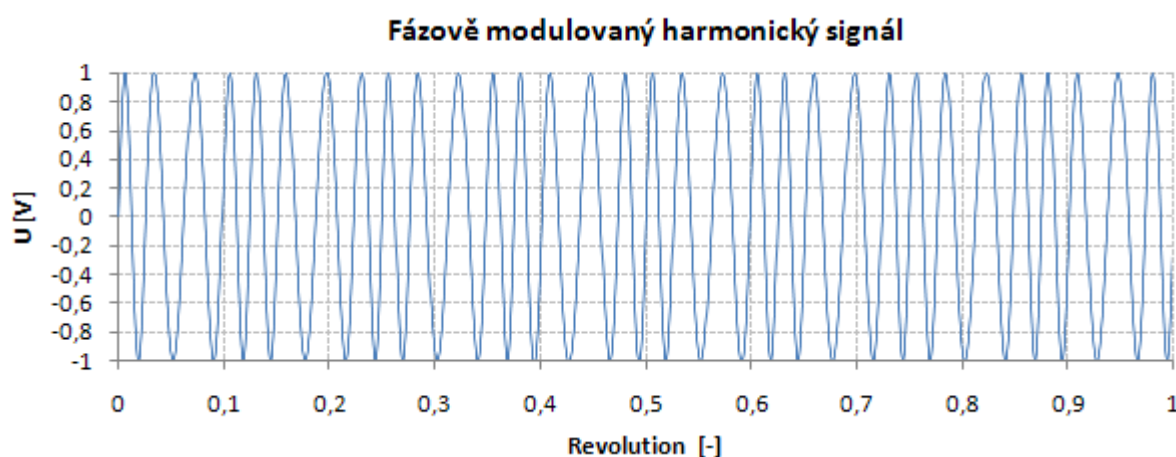
Výběr je proveden tak, že se po Fourierově transformaci (dále jen FT) impulzního signálu vynulují všechny složky spektra s výjimkou nosné složky a jejího postranního pásma. To je určeno frekvenční vzdáleností dalších složek spektra, které mohou být považovány za nosné složky. V případě impulzního signálu je to tedy druhá harmonická složka ke složce o základní frekvenci. Je velmi vhodné, aby se šířka horního a dolního pásma rovnala polovině základní frekvence. Zde počet složek postranního pásma určuje při malém indexu fázové

modulace nejvyšší harmonickou složku demodulovaného signálu. Pokud je tedy za jednu otáčku sejmuto K impulsů, pak nejvyšší harmonická demodulovaného není vyšší než $K/2$.



Obr. 2.10 Spektrum impulzního a harmonického signálu

Po vynulování nadbytečných složek je signál za pomoci inverzní FT převeden zpět do časové oblasti. Příklad fázově modulovaného signálu upraveného tímto druhem filtrace je prezentován na obr. 2.11. Je také dobré vědět, že pro nástroj FFT je výhodné, aby jednomu otočení odpovídal počet vzorků o velikosti mocniny dvou. [Tůma 2002]



Obr. 2.11 Fázově modulovaný harmonický signál

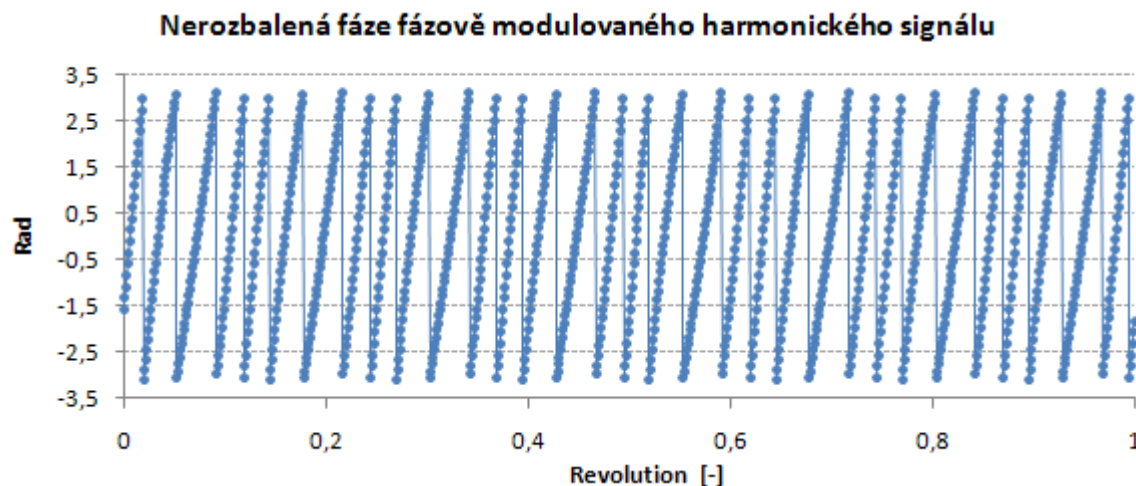
Základen nejen pro fázovou, ale i pro amplitudovou demodulaci signálu je teorie analytického signálu. K danému signálu $x(t)$ tedy náleží analytický signál $z(t)$, který je vytvořen podle následujícího vzorce:

$$z(t) = x(t) + jy(t) = |z(t)| \exp(j\varphi(t)) \quad (2.7)$$

Zde $x(t)$ představuje námi naměřený signál a $y(t)$ jeho Hilbertovu transformaci. Pro vzorkovaný signál $x(t)$ ve tvaru posloupnosti x_i , $i = 0, 1, 2, \dots, N-1$, mají také $y(t)$ a $z(t)$ tvar

posloupnosti délky N . Pokud Fourierovu transformaci reálného signálu označíme jako X_i , $i = 0, 1, 2, \dots, N-1$, je FT jeho imaginární části dána vztahem:

$$Y_i = -j \operatorname{sign}\left(\frac{N}{2} - i\right) X_i, \quad i = 0, 1, 2, \dots, N-1. \quad (2.8)$$



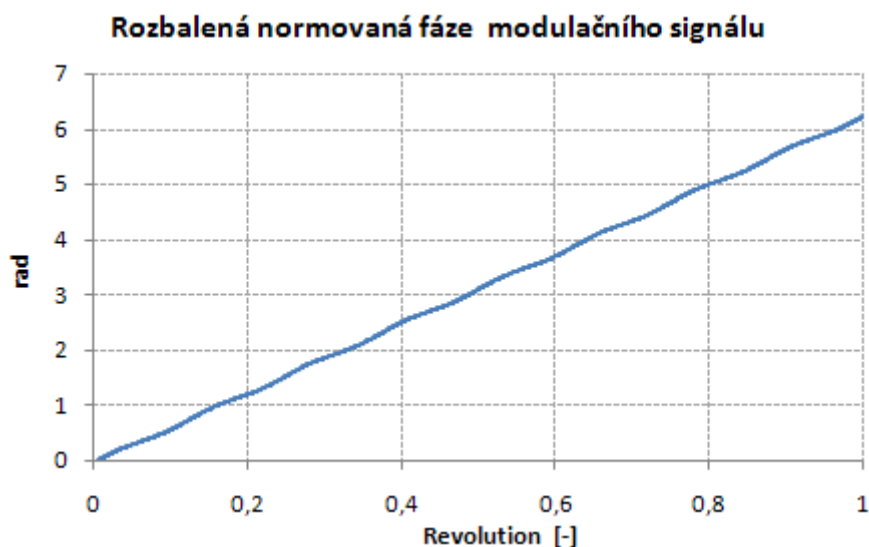
Obr. 2.12 Nerozbalená fáze fázově modulovaného harmonického signálu

Zpětnou FT posloupnosti Y_i , $i = 0, 1, 2, \dots, N-1$ poté dostane imaginární část analytického signálu. Vzorkovaný analytický signál je tvořen posloupností komplexních čísel, u kterých lze vypočítat absolutní hodnotu a fázi. Obálku signálu ve významu amplitudového modulačního signálu představuje absolutní hodnota:

$$|z_i| = \sqrt{x_i^2 + y_i^2}, \quad i = 0, 1, 2, \dots, N-1 \quad (2.9)$$

Pomocí funkce arkustangens lze z individuálních hodnot reálné a imaginární složky určit hlavní hodnoty fáze a to v intervalech od $-\pi$ do π .

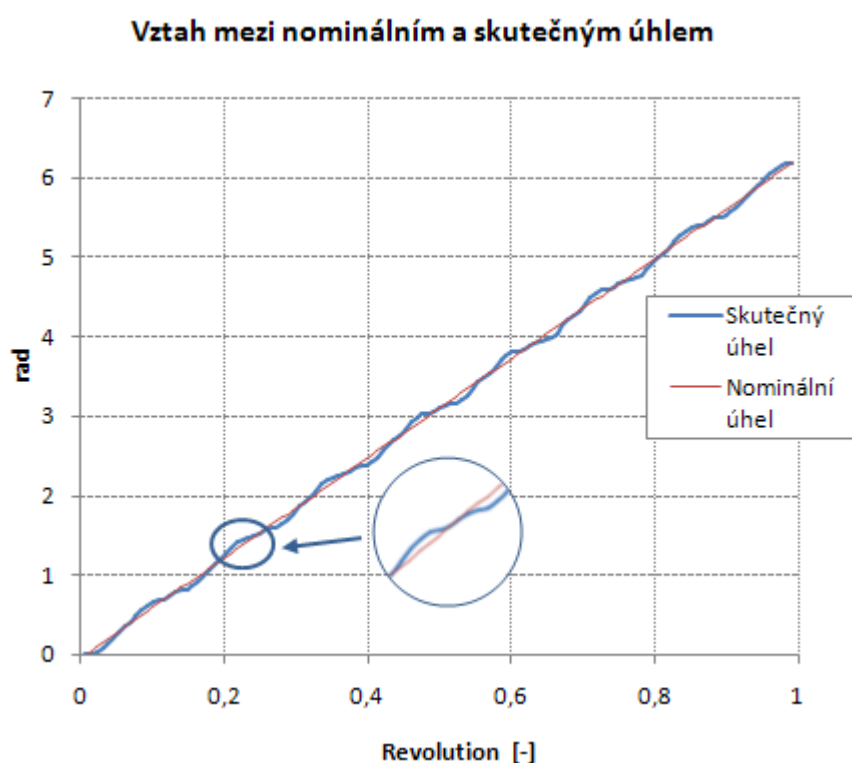
$$\operatorname{ARG}\{z_i\} = \arctan\left(\frac{y_i}{x_i}\right), \quad i = 0, 1, 2, \dots, N-1 \quad (2.10)$$



Obr. 2.13 Rozbalená a normovaná fáze modulačního signálu

Závislost hlavních hodnot fáze na čase obsahuje při dosažení hodnoty $\pm\pi$ body nespojitosti. Jedná se tedy o tzv. zabalenou (wrap) fázi, jejíž ukázka je presentována na obr. 2.12. Pokud však posuneme hlavní body fáze o celočíselný násobek π , získáme fázi (unwrap) rozbalenou. Algoritmus tohoto posunu se opírá o vyhodnocení rozdílu $ARG\{z_i\} - ARG\{z_{i-1}\}$ vzhledem k $2\pi - \varepsilon$. Volba velikosti koeficientu ε zde rozhoduje o tom, které skoky v průběhu fáze budou kompenzovány hodnotou $\pm 2\pi$. [Tůma 2002]

Při rozbalení se změní časový průběh fáze na rozsah v intervalu od 0 do násobku 2π . Ten je roven počtu impulsů za otáčku. Pro interpretaci výsledků je tedy vhodné normalizovat rozsah fáze na interval od 0 do 2π . Názorná ukázka rozbalené normované fáze je presentována na obr. 2.13.



Obr. 2.14 Vztah mezi nominálním a skutečným úhlem

Odstraněním lineárně narůstající složky v rozbalené normované fázi, která odpovídá konstantní úhlové rychlosti, získáme průběh odchylek od lineární funkce $\omega_o t$. Názorná ukázka tohoto průběhu, jejíž fyzikální rozměr je úhlová výchylka, je presentována na obr. 2.15. Pro hodnocení okamžité rychlosti je nutné úhlovou výchylku derivovat podle času. A pro určení zrychlení je potřeba dvojnásobné derivace. Tyto tři veličiny jsou charakterizovány následujícími vzorci. [Tůma 2002]

Úhel:

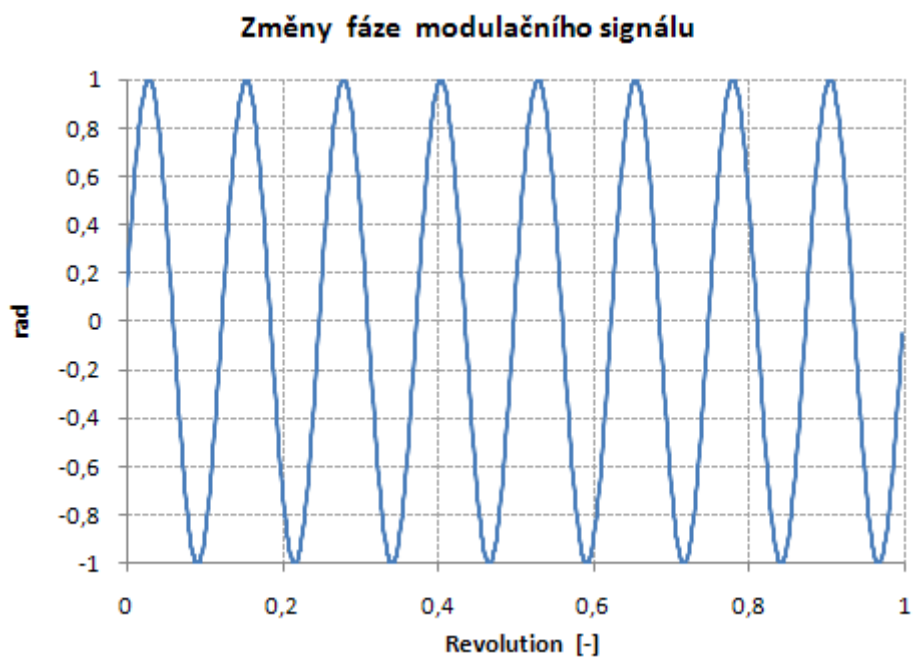
$$\varphi(t) = \frac{2\pi t}{T_o} + \Delta\varphi(t) = \omega_o t + \Delta\varphi(t) \quad (2.11)$$

Úhlová rychlost:

$$\omega(t) = \frac{d\varphi(t)}{dt} = \frac{2\pi}{T_o} + \frac{d\Delta\varphi(t)}{dt} = \omega_o + \Delta\omega(t) \quad (2.12)$$

Úhlové zrychlení:

$$\varepsilon(t) = \frac{d\omega(t)}{dt} = 0 + \frac{d\Delta\omega(t)}{dt} = \Delta\varepsilon(t) \quad (2.13)$$



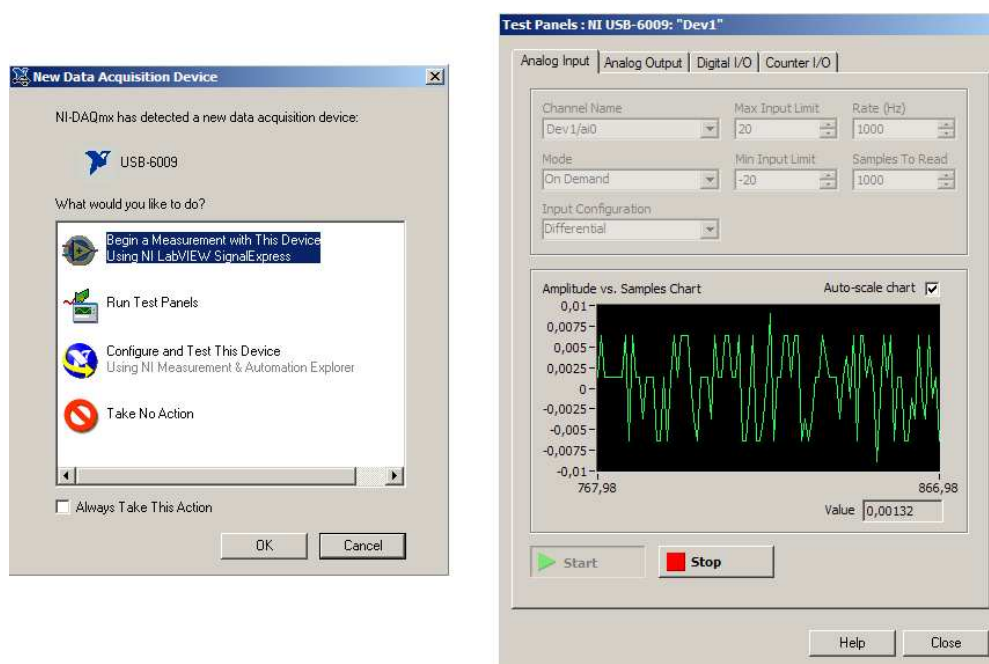
Obr. 2.15 Změny fáze modulačního signálu

3 Softwarové zajištění měřené úlohy

Technické prostředky již máme navrženy. Dostáváme se tedy k další velice podstatné části, která je věnována softwaru pro sběr a zpracování naměřených dat. Zde se opět zaměříme na popis karty NI USB-6009, tentokrát z pohledu softwarového vybavení. Další kapitoly již budou věnovány vývoji jednotlivých prvků vlastní aplikace zahrnující, sběr dat z měřicí karty a dále zapracování signálu pomocí digitálních filtrů. Sem spadá například Hilbertova transformace, horní, dolní a pásmová propust. Dále nástroj Tachometer, který slouží k učení nominální úhlové rychlosti a nástroj Autospectrum pro zobrazení frekvenčních spekter signálu.

3.1 Softwarová výbava NI USB - 6009

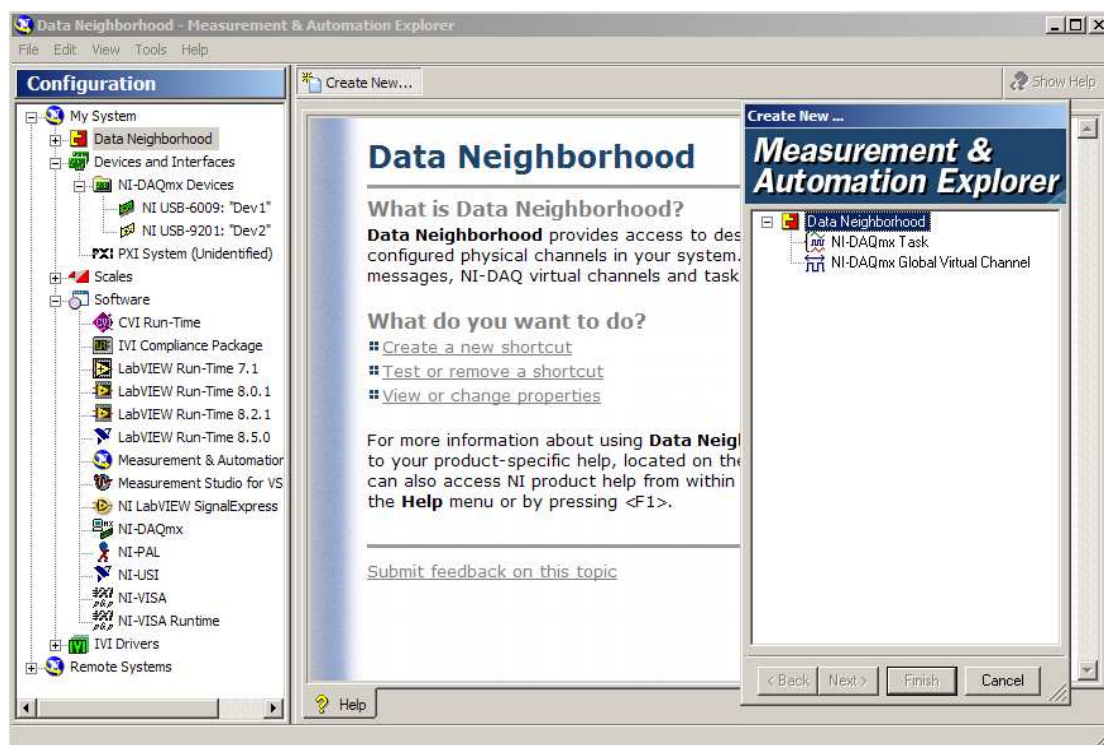
Se zakoupením karty NI USB-6009 obdržíte nejen podrobný návod pro použití, ale i jednoduché vzorové aplikace, vlastní měřicí software a mnoho dalších užitečných nástrojů pro vývoj vlastních aplikací. Tento software najdete pod souhrnným názvem NI-DAQmx na doprovodných CD dodávaných společně s touto kartou. Nejnovější aktualizace je pak možné získat na webových stránkách výrobce. NI-DAQmx je velice široký pojem, bez kterého se při práci s měřicími kartami od firmy National Instruments neobejdeme. Pojďme si tedy přiblížit některé jeho části.



Obr. 3.1 Testovací panel karty

Pokud již máme NI-DAQmx nainstalováno a připojíme propojovací USB kabel k počítači, zobrazí se uvítací dialog s několika možnostmi. Pokud kartu připojujeme poprvé, Diagnostika spalovacích motorů na základě měření kolísání jejich úhlové rychlosti

nebo jen chceme vyzkoušet její funkčnost, je dobré zvolit možnost „Run Test Panels”. Ten umožňuje otestovat jednotlivé vstupy a výstupy karty. Ukázka testovacího panelu je presentována na obr. 3.1.



Obr. 3.2 Measurement and Automation Explorer

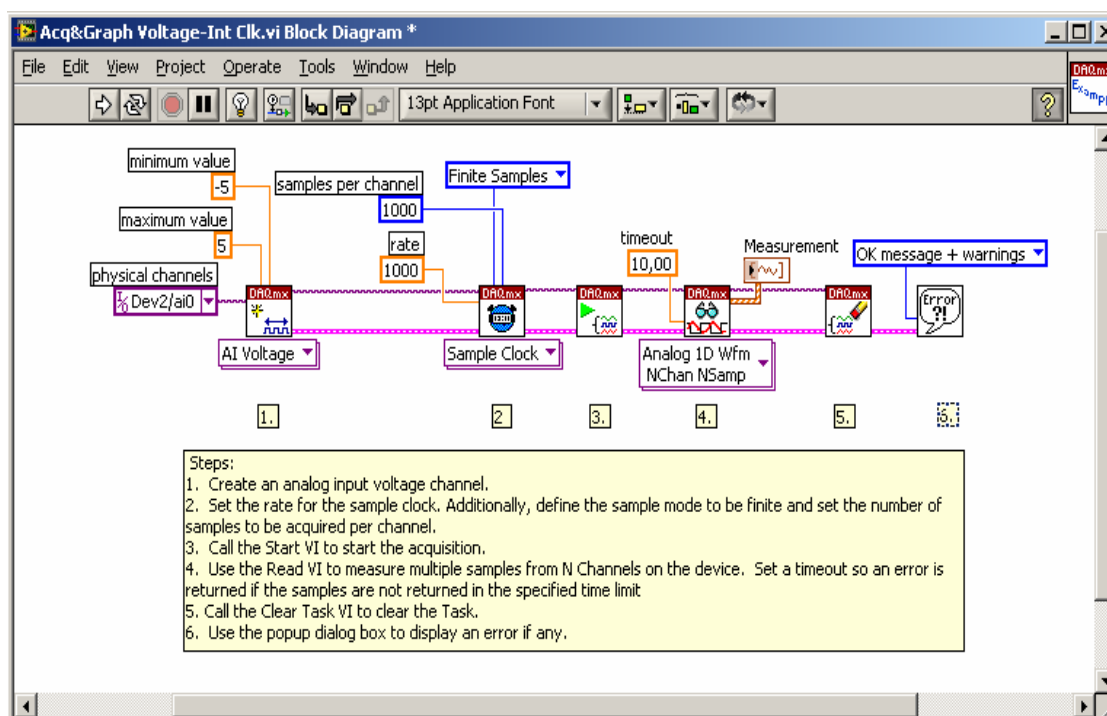
Další z nabídky je program, který je nazván Measurement and Automation Explorer, kterému se také s oblibou zkráceně říká MAX. Tento program je určen pro správu veškerých připojených zařízení od firmy National Instruments. Je zde tedy možné měnit nastavení všech připojených zařízení, vytvářet globální virtuální kanály a tasky, vytvářet vlastní měřítka, konfigurovat TEDS a mnoho dalšího. Velice užitečnou funkcí je i možnost vytvoření virtuálního zařízení, které nahrazuje reálné při vývoji měřicích aplikací. Tato funkce ovšem nezahrnuje námi vybraný typ karty. Ukázka tohoto programu je presentována na obr. 3.2.

Dále je vhodné, seznámit se s prostředím, ve kterých je možné vytvářet aplikace pro měření a následné zpracování dat. Lze sem zahrnout:

- LabVIEW
- MATLAB
- Microsoft Visual studio

LabVIEW je programovací a vývojové prostředí vyvinuté firmou National Instruments. Jedná se o grafické, blokově orientované prostředí, které je vhodné nejen k programování systémů pro měření a analýzu signálů, řízení a vizualizaci technologických procesů různé složitosti, ale také k programování velice složitých systémů. Pro tvorbu

aplikací je zde mnoho ovládacích prvků, seřazených přehledně v knihovnách. Najdeme zde prvky počínaje ovládacími a zobrazovacími (Grafy, knoflíky, diody, tabulky, atd.)



Obr. 3.3 Příklad aplikace ve vývojovém prostředí LabVIEW

přes matematické a logické funkce, funkce pro sběr a zápis dat a přístup do souboru, až po nástroje pokročilé analýzy. Toto shrnutí bylo velice stručné, neboť mimo knihoven obsažených v základní konfiguraci, lze LabView rozšířit o spoustu dalších knihoven z různých oblastí (obdoba toolboxů v Matlab). Velkou výhodou je i široká podpora operačních systému. Je možné vyvíjet aplikace pro UNIX, Windows (2000,XP,Vista), Linux i Mac. Ukázka tohoto programu je prezentována na obr. 3.3.

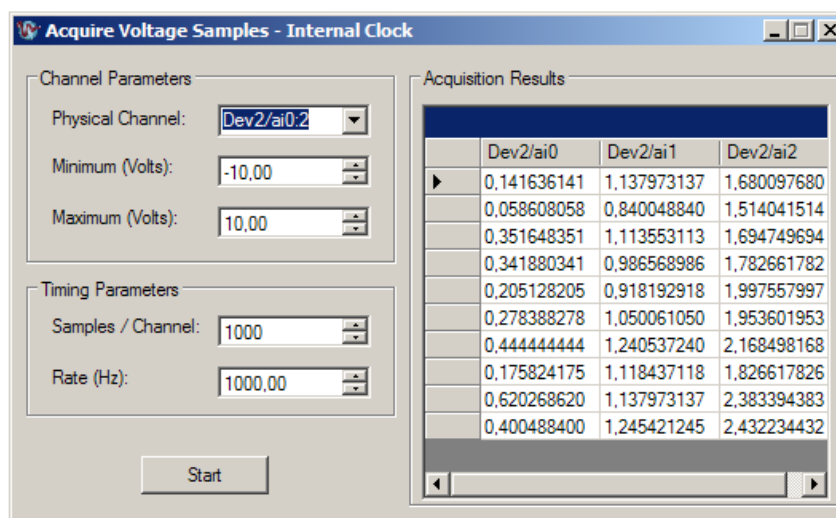
Dalším významným prostředím, kde lze vytvářet aplikace pro čtení a zpracování dat z měřicích karet nejen od firmy National Instruments je Matlab od firmy The MathWorks. Za tímto účelem se zde nachází Data acquisition toolbox. Tento toolbox umožňuje uživateli nastavení měření, přístupem k zabudovaným funkcím hardware a propojení s dalšími zabudovanými analytickými nástroji Matlabu a jeho toolboxů. Díky tomu můžeme naměřená data rychle analyzovat, zviditelnit, nebo ukládat pro další zpracování. Umožňuje tedy uživateli využít Matlab jako jednotné prostředí pro měření, analýzu a zpracování dat. Data acquisition toolbox podporuje velké množství měřicích karet, jejichž seznam je možné najít na webových stránkách výrobce spolu s nejnížší verzí toolboxu, od kterého byly implementovány. Karta NI USB-6009 je dostupná od verze 2.8.

Pokud chceme vytvořit novou uživatelskou aplikaci bez použití předchozích prostředí, nebo jen implementovat měřicí hardware do aplikací již existujících, nabízí National

Instruments prostřednictvím NI-DAQmx řešení v podobě přeprogramovaných knihoven funkcí. Při tvorbě vlastních aplikací se dále můžeme opřít i o vzorové aplikace, jejichž rozdělení naznačují okruh podporovaných programovacích jazyků, mezi které patří:

- ANSI C
- Microsoft Visual C++
- Microsoft C# .NET
- Microsoft Visual Basic .NET
- Microsoft Visual Basic 6.0

Další rozdělení vzorových úloh je například podle druhu měření, nebo měřené veličiny. Prostudovány byly veškeré vzorové příklady, ale vzhledem k povaze měřené úlohy byla největší pozornost věnována úlohám na měření napětí, v programových jazycích ANSI C a Microsoft Visual Basic .NET. Další podklady pro programování, byly čerpány ze stránek National Instruments, na kterých je pro vývoj vlastních aplikací věnován speciální prostor pod názvem „NI Developer Zone“. Ze studia jednotlivých podkladů vyplynulo, že pro zvládnutí podstatné většiny aplikací, vystačíme se základními deseti funkcemi. Ty budou popsány v následující kapitole. Jako ukázkou vzorového příkladu jsem vybral příklad, ze kterého jsem vycházel. Jeho grafické zpracování je prezentováno na obr. 3.4.



Obr. 3.4 Vzorová aplikace pro sběr dat

3.2 Sběr naměřených dat

Jak již bylo řečeno, ze studia vzorových příkladů, lokální nápovědy i článků na internetu vyplynulo deset základních funkcí pro založení vlastní aplikace. Z těchto funkcí nebyla v naší aplikaci použita pouze funkce pro nastavení triggru, který pro naše měření nebude potřebný. Tyto a další funkce knihovny NI - DAQmx .NET použité v naší aplikaci budou popsány v této kapitole.



Obr. 3.5 Postup tvorby algoritmu pro sběr dat

První krok při tvorbě měřicí aplikace je vytvořit task. Ten reprezentuje měřicí úlohu, která se skládá z jednoho nebo více virtuálních kanálů s časováním, spouštěmi a dalšími vlastnostmi. Nastavení tasku je možné uložit a použít jej i v jiných aplikacích. Konstruktor pro třídu „Task“ má dvě přetížené varianty. První, implicitní byla využita v naší aplikaci.

```
myTask = New Task()
```

Druhá varianta má jeden parametr typu string, kterým přidělujeme tasku pojmenování. Po vytvoření tasku následuje vytvoření virtuálních kanálů pro vstup či výstup dat z karty. Většina funkcí a tříd, které jsou součástí knihovny NI-DAQmx .NET, jsou také součástí třídy task. Je tomu tak i u tříd virtuálních kanálů, kterých je celkem šest:

- AIChannel (třída pro analogové vstupní signály)
- AOChannel (třída pro analogové výstupní signály)
- DIChannel (třída pro digitální vstupní signály)
- DOChannel (třída pro digitální výstupní signály)
- CIChannel (třída pro vstup čítače)
- COChannel (třída pro výstup čítače)

V naší aplikaci je využita třída pro analogové vstupní signály `AChannel`. Následuje ukázka funkce pro vytvoření virtuálního kanálu:

```
myTask.AChannels.CreateVoltageChannel(CardSet.PhysicalChannel, "",
AITerminalConfiguration.Differential, -CardSet.Ranges_
, CardSet.Ranges, AIVoltageUnits.Volts)
```

`CardSet.PhysicalChannel` je řetězec reprezentující reálné vstupy a výstupy zařízení. Existují přitom dva formáty jak daný řetězec zapsat. Prvním je řetězec `"Dev1/ai0:2"`, který zahrnuje vstupy `ai0`, `ai1` a `ai2` a druhým `"Dev1/ai0, Dev1/ai2"`, který na rozdíl od předchozího řetězce zahrnuje pouze vstupy `ai0` a `ai2`. `Dev1` je v tomto řetězci název zařízení, který mu je přidělen prostřednictvím MAX. `CardSet.Ranges` je parametr typu `double` pomocí kterého nastavujeme rozsah měření. Jejich rozmezí udává tabulka 2.1. `AITerminalConfiguration.Differential` je výčtový datový typ pomocí kterého nastavíme režim čtení na „Single-ended“, nebo na „Differential“. Rozdíl mezi těmito režimy je již popsán v předchozí kapitole. `AIVoltageUnits.Volts` je též výčtovým datovým typem, pomocí kterého se nastavují jednotky měření. Pro měření je důležité také nastavit vlastnosti vzorkování. K tomuto účelu slouží funkce `ConfigureSampleClock`.

```
myTask.Timing.ConfigureSampleClock("", CardSet.SampleRate,
SampleClockActiveEdge.Rising, SampleQuantityMode.FiniteSamples_
, CardSet.SampPerChan)
```

První parametr udává zdroj časového signálu. Tento parametr je v našem případě prázdný řetězec. Používáme tedy vzorkovací hodiny měřicí karty. Při použití jiného časového zdroje by řetězec vypadal následovně `"/Dev1/PFI0"`. `CardSet.SampleRate` je parametr typu `integer`, udávající vzorkovací frekvenci. `SampleClockActiveEdge.Rising` je výčtovým typem prostřednictvím kterého se nastavuje spouštěcí hrana. Zvolena byla hrana náběžná, neboť spádovou karta NI USB-6009 nepodporuje.

`SampleQuantityMode` tento parametr je opět výčtovým typem u kterého je možné volit ze tří možností. Při volbě `FiniteSamples` se po naměření požadovaného počtu vzorků ukončí měření. Při volbě `ContinuousSamples` měření probíhá průběžně, přičemž po každém dosažení nastaveného počtu vzorků, jsou data odeslána programu přes funkci `ReadMultiSample`. Obě předchozí nastavení využívají buffer. Opakem je tomu u poslední možnosti `HardwareTimedSinglePoint`, kde je čtení a zápis prováděn po jednom vzorku. Posledním parametrem `CardSet.SampPerChan` nastavíme počet požadovaných vzorků. Po nastavení všech potřebných vlastností tasku provedeme jeho verifikaci.

```
' Verifikace Tasku
myTask.Control(TaskAction.Verify)
```

Předání naměřených dat z měřicí karty je v naší aplikaci provedeno následujícími dvěma postupy. Prvnímu předchází použití výše uvedených funkcí, bez jakékoliv změny parametrů. Následně je pro sběr dat z měřicí karty použita funkce `ReadMultiSample` nově vytvořeného objektu `reader`.

```
' Cteni dat
reader = New AnalogMultiChannelReader(myTask.Stream)
Measur.AcquiredData = reader.ReadMultiSample(CardSet.SampPerChan).
```

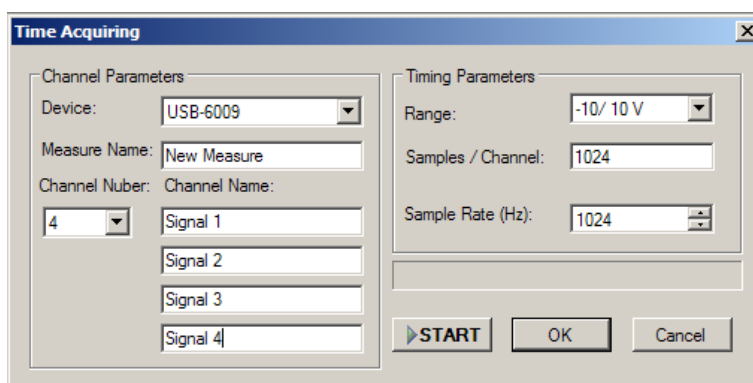
Požadovaná data jsou při tomto postupu naměřena a uložena v bufferu měřicí karty jako celek, je tedy vhodné takto získávat data o menší délce. Námi zvolená délka je `OPTIMAL_SAMPLE = 512`, která však zdaleka není limitem bufferu zvolené měřicí karty. Po naměření dat a následném předání proměnné v aplikaci měřicí úloha končí.

Při měření většího počtu vzorků využijeme druhého postupu. Zde jsou data brány v menších dávkách a postupně ukládány do pomocné proměnné v aplikaci. Celý proces se rekurzivně opakuje, dokud není naměřen potřebný počet vzorků, nebo dokud není zrušen task. Tomuto postupu opět předchází výše uvedené funkce. Malá změna je provedena pouze u funkce `ConfigureSampleClock`. Zde je parametr `SampleQuantityMode` nastaven na hodnotu `SampleQuantityMode.ContinuousSamples`. Pro sběr dat jsou u tohoto postupu použity následující funkce.

```
reader.SynchronizeCallbacks = True
analogCallback = New AsyncCallback(AddressOf AnalogCallbackGetData)
reader.BeginReadMultiSample(Convert.ToInt32(OPTIMAL_SAMPLE), _
analogCallback, myTask)
```

Knihovna `NI-DAQmx.NET` obsahuje nejen funkce pro sběr dat, ale i mnoho dalších funkcí umožňující získávat informace o připojených zařízeních. Velice užitečnou se jeví zejména níže uvedená funkce, sloužící ke zjištění názvů všech aktuálně připojených měřicích zařízení firmy National Instruments.

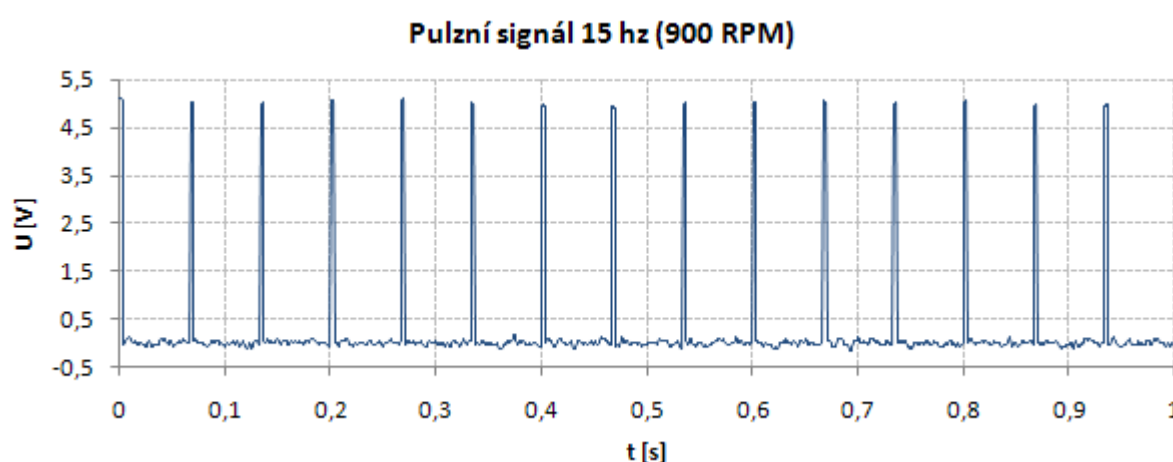
```
Dim DAQdevices() As String = DaqSystem.Local.Devices.ToArray()
```



Obr. 3.6 Dialogové okno pro nastavení karty

Algoritmus pro měření dat z karty je v aplikaci použit na několika místech. Jedním z nich je i formulář, sloužící k měření signálů určité časové délky viz obr. 3.6.

Logickým krokem po vytvoření algoritmu pro sběr dat z měřicí karty, bylo otestování jeho funkčnosti. Prostřednictvím programu Signal Analyser a zvukového zařízení počítače byly na měřicí kartu přiváděny různé signály. Měřením těchto signálů naší aplikací a následné porovnání se signálem původním potvrdilo plnou funkčnost této části aplikace. Jako ukázka naměřeného signálu byl zvolen zašuměný pulzní signál s frekvencí 15 Hz, jenž odpovídá 900 RPM (volnoběžné otáčky osobního automobilu). Tento signál bude předmětem zpracování při kontrole nástroje Tachometer.



Obr. 3.7 Kontrolní pulzní signál odpovídající volnoběžným otáčkám

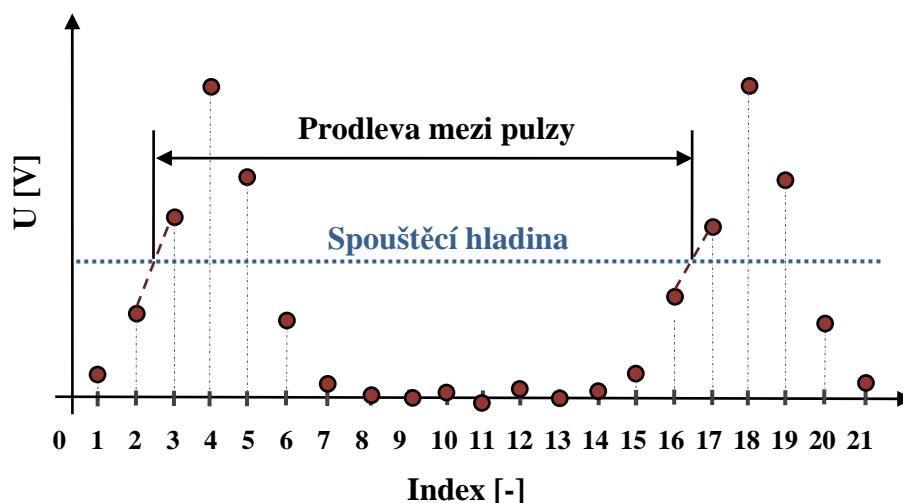
3.3 Tachometer - nominální rychlost otáčení

V předchozí kapitole jsme se seznámili s funkcemi, pomocí kterých jsme schopni získávat naměřená data z měřicí karty. Tyto data však pro nás nemají potřebnou vypovídající hodnotu, proto je nutné jejich další zpracování. Za tímto účelem bude naše aplikace poskytovat několik nástrojů. První z těchto nástrojů slouží pro získání nominální rychlosti otáčení a nazývá se Tachometer.

Předmětem zpracování tohoto nástroje je pulzní signál, který bude generován tachosondou, IRC snímačem, nebo v případě spalovacího motoru snímačem umístěným na vačkovém hřídeli. Hodnocení rychlosti otáčení vychází z určení časové prodlevy mezi jednotlivými impulzy v tomto signálu (viz obr. 3.8 a obr. 3.7).

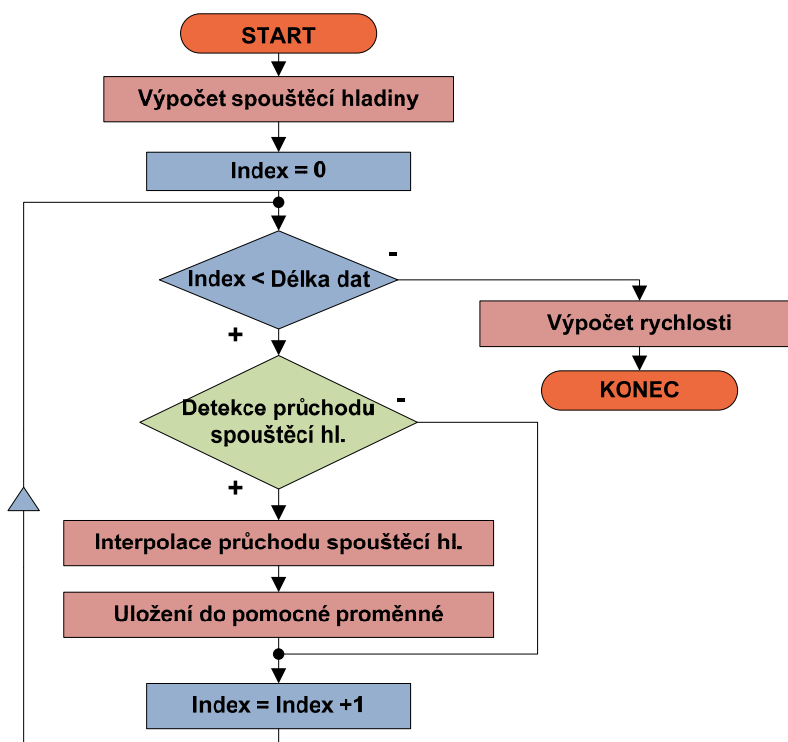
Pro detekci impulsu algoritmem je zapotřebí nejdříve určit spouštěcí hladinu, kterou lze nastavit v dialogovém okně pro tento nástroj za pomoci položky Level. Jedná se o procenta z maximální hodnoty naměřeného signálu. Časová prodleva mezi impulzy je tedy určena dvěma průchody přes spouštěcí hladinu. Způsob průchodu touto hladinou je možné

změnit pomocí položky Sloupe na positive, pokud chceme detekovat pulz na náběžné hraně, nebo negativ pro detekci pulzu na hraně spádové viz obr. 3.10.



Obr. 3.8 Ukázka vzorkovaného pulzního signálu (detekce impulsu algoritmem)

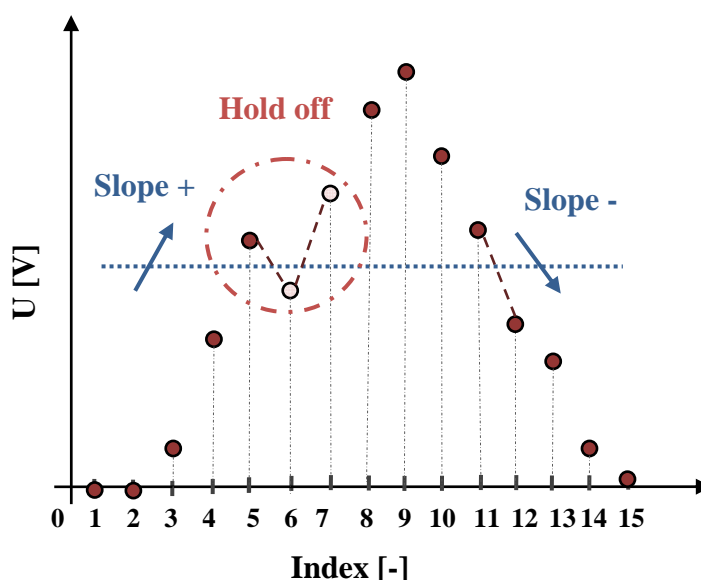
Časová prodleva mezi dvěma náběžnými hranami je pak dána podílem počtu vzorků a vzorkovací frekvence. Pro vzorkovací frekvenci tedy platí nepřímá úměra mezi počtem impulsů generovaných za otáčku a relativní přesností měření délky časového intervalu. Přesnost vyhodnocení je přitom dána poměrem frekvence první harmonické signálu a vzorkovací frekvence.



Obr. 3.9 Struktura algoritmu pro získání nominální rychlosti otáčení

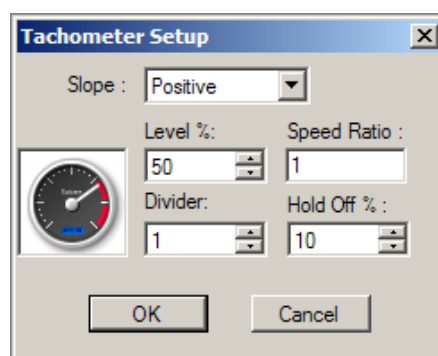
Neurčitost z titulu vzorkovací periody je v našem algoritmu snížena interpolací okamžiku průchodu signálu spouštěcí hladinou. Chyba odečtu se tak sníží asi padesátkrát. Je

zřejmé, že dobu mezi impulzy v počtu jeden za otáčku lze vyhodnotit velmi přesně, zatím co pro stovky impulzů za otáčku připadá často na jednu periodu impulzů jen několik vzorků signálu. Mimo dříve popsané vlastnosti nástroj Tachometer disponuje ještě Speed Ratio, jenž slouží k zadání převodového poměru a dále funkci Hold Off. Zde se jedná o procentuální hodnotu z časové prodlevy mezi jednotlivými pulzy, na které je vypnuta detekce průchodu spouštěcí hladinou. Slouží tedy jako prevence před šumem kolem spouštěcí hladiny.



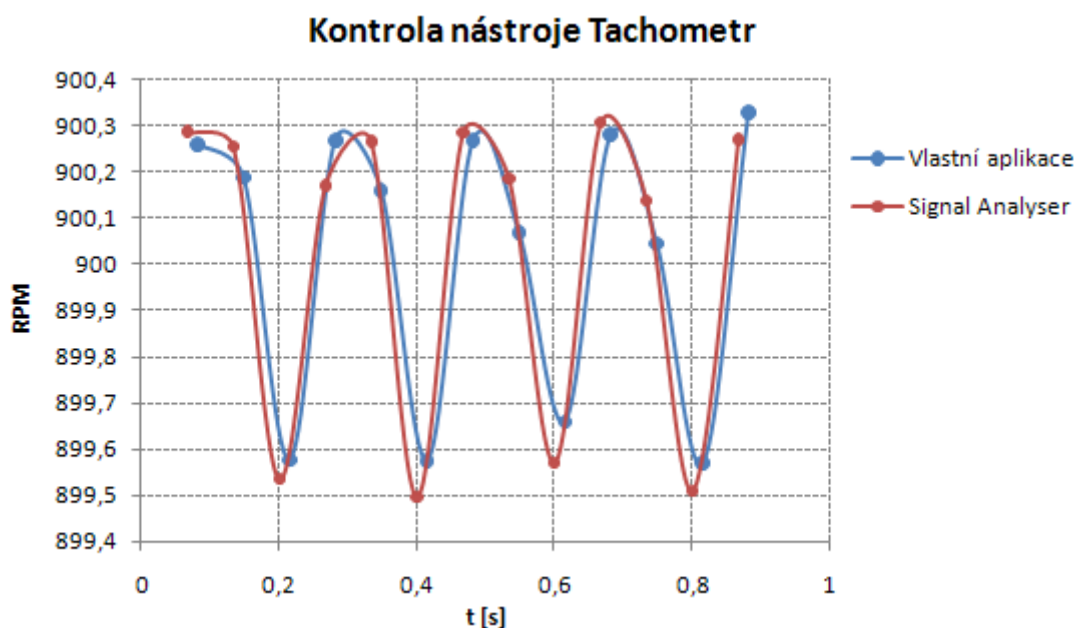
Obr. 3.10 Ukázka vlastnosti nástroje Tachometer na samostatném pulzu

Poslední vlastnost Divider využijeme, pokud je generováno více, než jeden pulz za otáčku. V tomto případě je zadán počet pulzů za otáčku. Dialogové okno pro nastavení vlastnosti nástroje Tachometer je prezentováno na obr. 3.11.



Obr. 3.11 Dialogové okno pro nastavení Nástroje Tachometer

Pro kontrolu tohoto nástroje byl na jeho vstup přiveden vygenerovaný zašuměný pulzní signál o frekvenci 15 Hz, který se svou frekvencí blíží volnoběžným otáčkám automobilu. Tento signál je prezentován na obr. 3.7. Výslednou nominální rychlost otáčení vlastní aplikace jsem následně porovnal s výsledky programu Signal Analyser. Toto srovnání je prezentováno na obr. 3.12.



Obr. 3.12 Porovnání výsledků vlastní aplikace s programem Signal Analyser

3.4 Autospectrum – frekvenční spektrum signálu

Jeden ze základních nástrojů využívaných při zpracování signálů je Autospectrum, který slouží k zobrazování frekvenčních spekter. Protože zobrazení frekvenčního spektra signálu, využívaných pro vyhodnocení rovnoměrnosti otáčení, může mít větší vypovídající hodnotu, než jeho časový průběh, byl tento nástroj vytvořen i v naší aplikaci. Struktura algoritmu zajišťující výpočet spektra je prezentován na obr. 3.13



Obr. 3.13 Struktura algoritmu zajišťující výpočet spektra

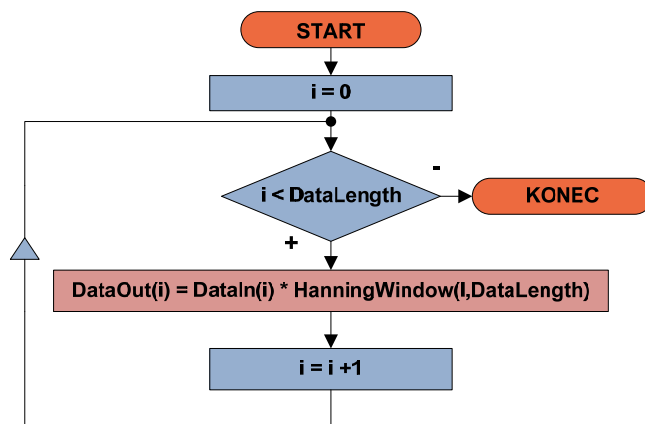
Jako první byla vytvořena časová okna. Do skupiny vytvořených oken patří:

- Rectangular
- Hanning
- Flat Top
- Kaiser Bessel

Pro názornou ukázkou bylo zvoleno okno Hanning, které je popsáno vztahem 3.1, kde k je index a n je délka výstupního vektoru FFTW:

$$w(k) = 1 - \cos\left(\frac{2\pi k}{n}\right) \quad (3.1)$$

```
Function HanningWindow(ByVal k As Long, _
                        ByVal Length As Long) As Double
    Return (1 - Cos((2 * PI * k) / Length))
End Function
```



Obr. 3.14 Algoritmus realizující aplikaci časových oken na vstupní signál

Pro výpočet diskretní Fourierovy transformace (dále jen DFT) byla použita komponenta založená na algoritmu the Fastest Fourier Transform in the West (dále jen FFTW). Tento algoritmus vyvinuli Matteo Frigo a Steven G. Johnson. Komponenta je schopna provádět výpočet DFT pro reálná i komplexní data o libovolné dimenzi a libovolné délce (včetně délky rovné prvočíslu). FFTW používá pro výpočet mnoho různých rutin, proto je výpočet rozdělen do dvou hlavních fází. V první fázi se hledá vhodný algoritmus pro vstupní data a vytváří se tzv. plán. Ten je závislý na délce těchto dat. Pro případ, kdy se provádí opakovaný výpočet se stejnou délkou transformace, není nutné plán rušit. Ve druhé fázi teprve probíhá samotný výpočet transformace s využitím vytvořeného plánu. Další velkou předností tohoto algoritmu je i vysoká rychlost výpočtu. Komponenta byla do programu zařazena teprve po odzkoušení na jednoduchých příkladech. Zde byly pomocí níže uvedené funkce generovány harmonické signály a po průchodu DFT kontrolovány výsledky.

```
Private Function Test(ByVal x As Double) As Double
    Return (AMPLITUDE * Sin(2 * PI * x / SAMPLERATE * FREQUENCY))
End Function
```

Výstupem této komponenty je pole, jehož délka je o jeden prvek větší, než u pole vstupního. V tomto poli jsou uloženy jak reálné tak i imaginární složky výsledné DFT. Jejich rozložení je prezentováno na obr. 3.15.



Obr. 3.15 Rozložení imaginárních a reálných složek ve výstupním poli komponenty

Výsledkem výpočtu DFT je dekompozice signálu na harmonické frekvenční složky. Při vstupním vektoru o délce n , je absolutní hodnota složky signálu n násobně větší než jeho amplituda a harmonická složka spektra připadá na frekvenční pásmo o šířce převrácené hodnoty doby měření t vstupním vektoru. Pro praxi je toto měřítko nepřehledné a proto byl výstup komponenty FFTW přepočítán do měřítka nového. Zde na ose x vystupuje frekvence v hertzech a pro amplitudu je možné zvolit tři varianty: [Tůma 2008]

- RMS (Efektivní hodnota)
- PWR (Celkový výkon signálu)
- PSD (Výkonová spektrální hustota)

Pro ukázkou je uvedena část kódu zabývající se výpočtem RMS. Výpočet efektivní hodnoty je dán vztahem 3.2, Kde x jsou hodnoty odpovídající výstupnímu vektoru FFTW.

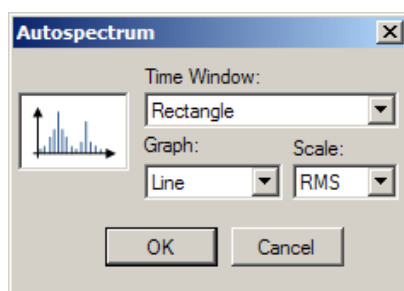
$$RMS(k) = \begin{cases} \frac{x_0}{n} & ; k = 0 \\ 2 \cdot \frac{|x_k|}{n \cdot \sqrt{2}} & ; k > 0 \end{cases} \quad (3.2)$$

```

.
.
Case Scale.RMS
  For k As Integer = 0 To UBound(DataIn)
    If k = 0 Then
      Ydata(k) = Abs(DataIn(k) / CDb1(NumOfSample))
    Else
      Ydata(k) = Abs((2 * DataIn(k)) / (Sqrt(2) * NumOfSample))
    End If
  Next
Case Scale.PWR
.
.

```

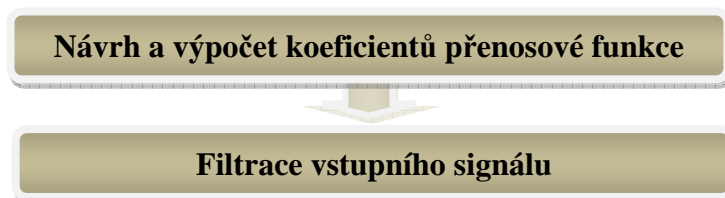
Pro nastavení nástroje Autospectrum bylo vytvořeno dialogové okno viz obr. 3.16.



Obr. 3.16 Dialogové okno pro nastavení Nástroje Autospectrum

3.5 Filter - digitální filtry

Dalším nástrojem pro zpracování signálu vytvořeným v naší aplikaci je Filters. Ten využívá digitálních filtrů pro odstranění uživatelem vybraných frekvenčních složek vstupního signálu. Struktura algoritmu zajišťující funkci tohoto nástroje prezentuje obr. 3.17.

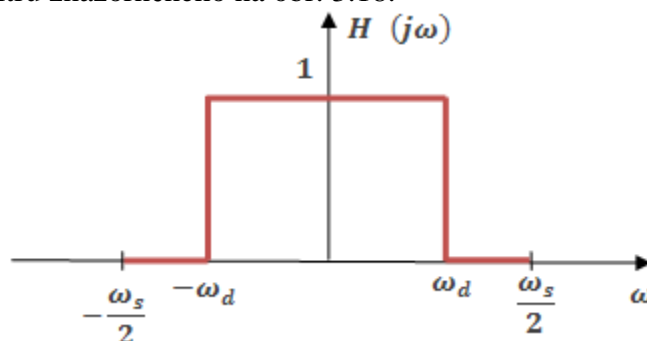


Obr. 3.17 Struktura algoritmu nástroje Filters

Rozeznáváme dva druhy digitálních filtrů, IIR s nekonečnou odezvou a FIR s konečnou odezvou, kterým byla dána v naší aplikaci přednost. Filtry typu FIR představují typ, jenž nemá analogii u analogových filtrů. Jsou určeny svou přenosovou funkcí, která má M – násobný pól v nule, jsou tedy nezávislé na koeficientech a vždy stabilní. Přenosová funkce FIR filtru má následující tvar: [Tůma 2009]

$$H(z) = b_0 + b_1 z^{-1} + \dots + b_{M-1} z^{-M+1} + b_M z^{-M} \quad (3.3)$$

Na ukázkou byl zvolen návrh dolnoproustného filtru, který vychází z ideálního dolnoproustného filtru znázorněného na obr. 3.18.



Obr. 3.18 Zadání ideálního dolnoproustného filtru

Nejdříve tedy potřebujeme získat koeficienty přenosové funkce. Zde vycházíme z frekvenční přenosové funkce dolnoproustného filtru. [Tůma 2009]

$$H(j\omega) = \begin{cases} 1; & -\omega_d \leq \omega \leq \omega_d \\ 0; & \frac{\omega_s}{2} > \omega > \omega_d, \quad -\frac{\omega_s}{2} < \omega < -\omega_d \end{cases} \quad (3.4)$$

Je také třeba uvažovat frekvenční omezení měření podle Shannon – Kotelníkova teorému, který vyjadřuje vztah 3.5, kde f_s označuje vzorkovací frekvenci a ω_s kmitočet odpovídající této frekvenci:

$$f_s = \omega_s / 2\pi \quad (3.5)$$

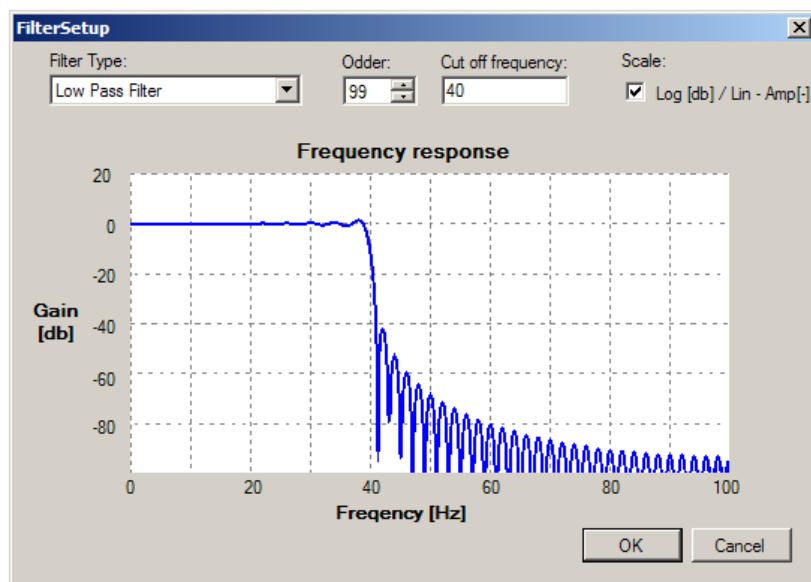
Zpětnou transformací frekvenční přenosové funkce pro omezený rozsah úhlové frekvence do poloviny kladné a záporné vzorkovací frekvence je tedy impulzní odezva a platí:

$$h_n = \frac{1}{2\pi} + \int_{-\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} H(j\omega) \exp(j\omega n T_s) d\omega = \frac{\sin\left(2\pi n \frac{f_d}{f_s}\right)}{\pi n}; \quad n = 0, \pm 1, \pm 2 \dots \quad (3.6)$$

Zde f_d označuje hraniční frekvenci dolnoproústného pásma, f_s označuje vzorkovací frekvenci a n je řád filtru.

Z předchozího vzorece je možné vyjádřit následující část programového kódu:

```
' Požadavek na řád filtru: Odder > 3 And (Odder Mod 2) <> 0
Public Sub LowPassFilt( ByVal Odder As Integer, ByVal f As Double, _
                        ByRef FilterCoef() As Double)
    ReDim FilterCoef((FilterOdder - 1))
    m =(FilterOdder - 1) \ 2
    Wcutoff = PI * f
    For n = -m To m
        If n = 0 Then
            FilterCoef(n + m) = Wcutoff / PI
        Else
            FilterCoef(n + m) = Sin(Wcutoff * n) / (PI * n)
        End If
    Next n
End Sub
```



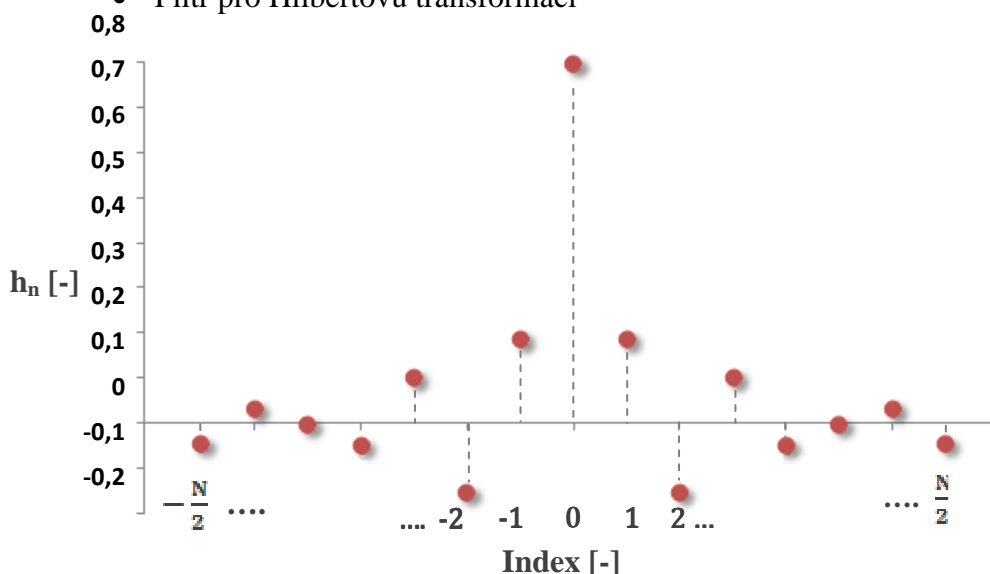
Obr. 3.19 Dialogové Okno pro nastavení nástroje Filters

Využíváme filtrů nekauzálních, což znamená, že odezva předchází vstupní impuls. Pro dodatečné zpracování naměřených signálu není nutné na kauzalitě filtru trvat, protože známe předcházející vstupní i výstupní vzorky. Rovněž vzhledem k zmenšující se absolutní hodnotě koeficientů a vzhledem k rostoucímu n je možné zkrátit odezvu na vybraný počet koeficientů

tak, že vznikne FIR filtr. Počet koeficientu je roven řádu filtru (`FilterOrder`). Podobným postupem jakou u filtru typu dolní propust, bylo postupováno i u dalších filtrů. [Tůma 2009]

V aplikaci jsou tedy vytvořeny tyto filtry:

- Dolní propust
- Horní propust
- Pásmová propust
- Filtr pro Hilbertovu transformaci



Obr. 3.20 Koeficienty přenosové funkce nekauzálního FIR filtru

Po návrhu koeficientů přenosové funkce filtru můžeme přejít procesu filtrace. Výstup nekauzálního FIR filtru je dán vztahem 3.7:

$$y_n = x_{n-\frac{N}{2}} \cdot h_{\frac{N}{2}} + x_{n-\frac{N}{2}+1} \cdot h_{\frac{N}{2}-1} + \dots + x_n \cdot h_0 + x_{n+1} \cdot h_{-1} + \dots + x_{n+\frac{N}{2}} \cdot h_{-\frac{N}{2}} \quad (3.7)$$

Zde je h přenosová funkce, x vstupní signál a y signál výstupní. Tento vztah lze zapsat pomocí následujícího programového kódu:

```

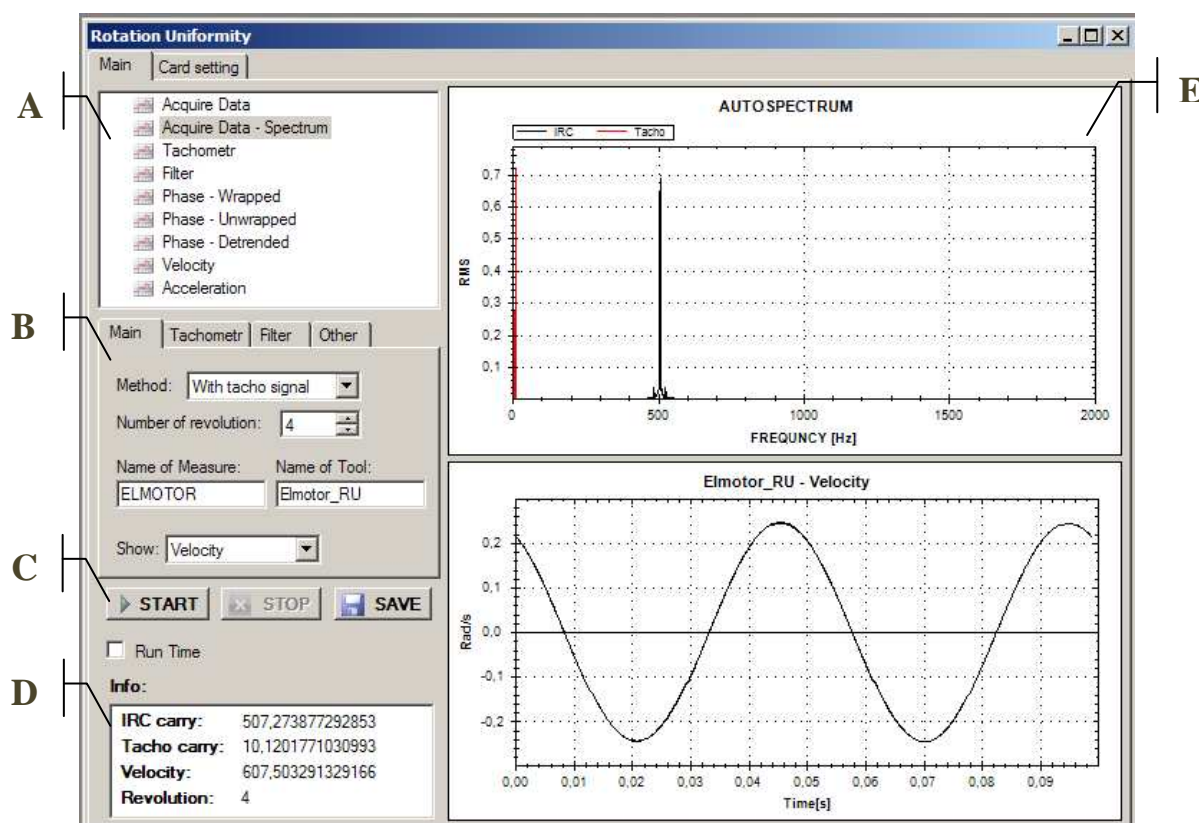
.
. Vystup filtru
Dim Ofset = UBound(FiltrCoef) \ 2
For VectorIndex As Integer = 0 To UBound(DataVector)
    Dim FilterIndex = UBound(FiltrCoef)
    For i As Integer = VectorIndex - Ofset To VectorIndex + Ofset
        If (i >= 0 And i <= UBound(DataVector)) Then
            OutVector(VectorIndex) += FiltrCoef(FilterIndex) * _
                                   DataVector(i)
        End If
        FilterIndex -= 1
    Next i
Next VectorIndex
.
.

```

3.6 Rotation uniformity - rovnoměrnost otáčení

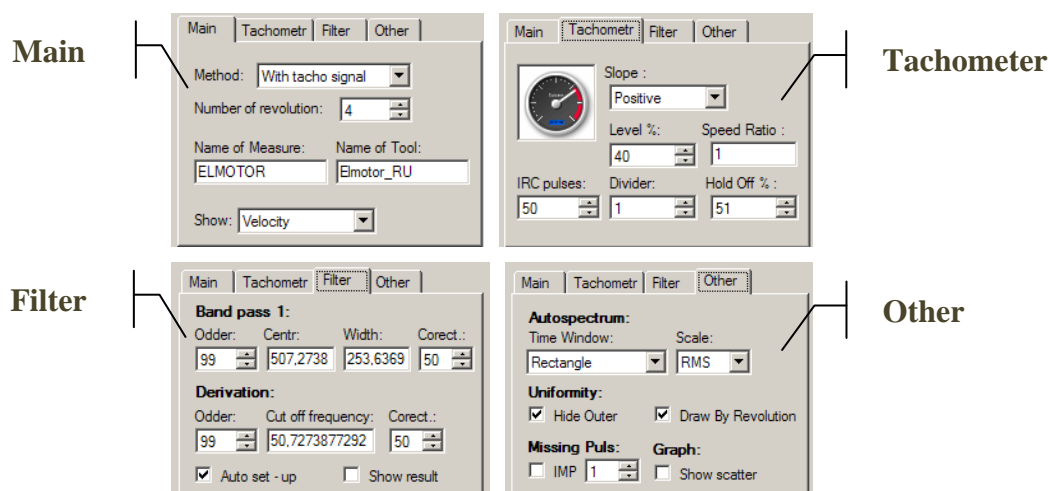
Jak již název napovídá, tento nástroj slouží k hodnocení rovnoměrnosti otáčení. Je založen na dříve popsané metodě fázové demodulace a využívá předešlých nástrojů tak, aby byly soustředěny na jednom místě všechny potřebné informace k vyhodnocení daného problému. Uživatel si tak kromě konečného výsledku může zobrazit i naměřená vstupní data, mezivýsledky jednotlivých kroků a samozřejmě je i zobrazení spekter všech výše uvedených signálů.

Předmětem zpracování tohoto nástroje je pulzní signál generovaný IRC snímačem. Ten podle typu vyprodukuje stovky, až tisíce pulzů za otáčku. V případě měření rovnoměrnosti otáčení spalovacího motoru, bude zdrojem IRC signálu snímač, umístěný na setrvačniku motoru. Dále bude tímto nástrojem zpracováván tachosignál, což je pulzní signál, který je generován tachosondou, IRC snímačem, nebo v případě měření rovnoměrnosti otáčení spalovacího motoru, snímačem umístěným na vačkovém hřídeli.



Obr. 3.21 Ukázka Formuláře nástroje Rotation uniformity – Záložka Main

- A) **Volba zobrazení signálu:** v části formuláře Rotation uniformity (viz obr. 3.21) označené písmenem A je umístěn ovládací prvek treeview, jehož položky nesou názvy jednotlivých signálů. Zvolením libovolné položky uživatel aktualizuje nastavení tohoto nástroje a vykreslí uživatelem zvolený signál.



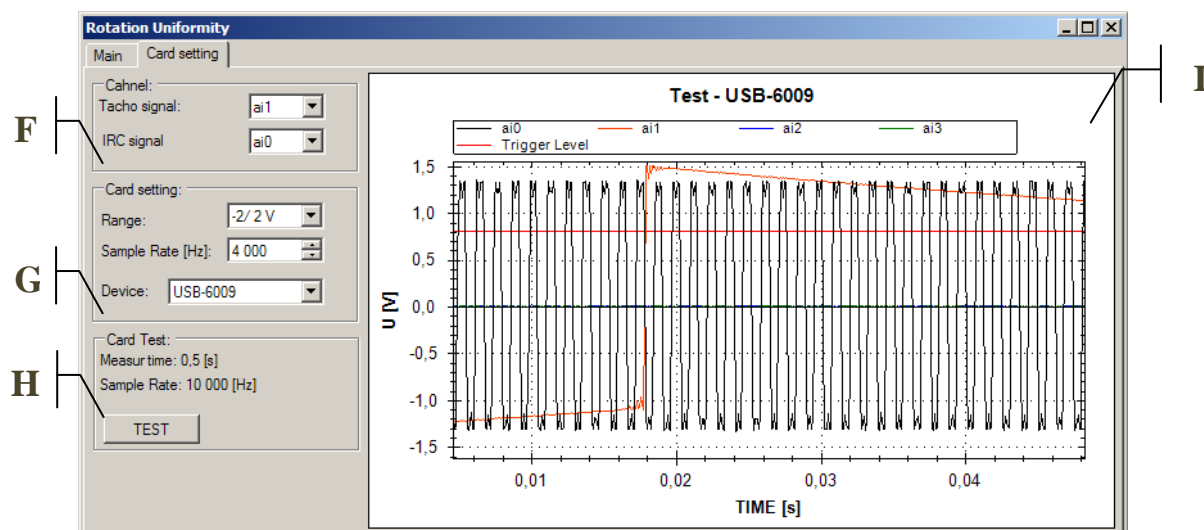
Obr. 3.22 Záložky pro nastavení Nástroje Rotation Uniformity

Tabulka 3.1 Parametry Nástroje Rotation uniformity

Main	
Number of revolution	Počet otáček, které požaduje uživatel naměřit
Name of Measure	Pojmenování měření v organizéru signálů MainPanel
Name of Tool	Pojmenování nástroje v organizéru nástrojů MainPanel
Show	Určuje, který z konečných výsledků bude zobrazen společně s mezivýsledkem
Tachometer	
Slope	Určuje detekci spádové, či náběžné hrany
Level	Procentuální hodnota z maxima signálu. Určuje spouštěcí hladinu
Speed Ratio	Převodový poměr
Divider	Počet pulzů za otáčku v tachosignálu
Hold off	Procentuální hodnota z časové prodlevy mezi pulzy, kdy je vypnuta detekce průchodu spouštěcí hladinou
IRC pulses	Zadáváme celkový počet pulzů v IRC signálu, včetně chybějících
Filter	
Odder	Řád filtru
Center	Střed propustného pásma Pásmového filtru. Hodnota by měla být rovna nosné složce ve spektru IRC signálu.
Width	Šířka propustného pásma Pásmového filtru. Hodnota by měla být rovna max. polovině položky Center
Correct.	Procentuální hodnota z nosné složky ve spektru IRC signálu. Slouží k automatickému výpočtu parametru filtru
Auto set - up	Vypnutí a zapnutí automatického nastavení filtrů
Show result	Zobrazí výsledné spektrum po filtraci
Other	
Time window	Časová okna viz kapitola Autospectrum – frekvenční spektrum signálu
Scale	Měřítka pro zobrazení spekter RMS, PWR, PSD
Hide outer	Skryje okrajové části signálu, které jsou znehodnoceny náběhem filtru
Draw by revolution	Vykreslení výsledků po otáčkách
IMP	Interpolace chybějících vzorků.
Show scatter	Vykreslí body v grafu

- B) Nastavení Nástroje:** v části formuláře Rotation uniformity (viz obr. 3.21) označené písmenem B je možné pomocí několika záložek nastavit parametry tohoto nástroje. Tyto záložky jsou prezentovány na obr. 3.22 a jejich jednotlivé položky a popis jejich peremetrů je uveden v tabulce 3.1.
- C) START, STOP, SAVE:** v části formuláře Rotation uniformity (viz obr. 3.21) označené písmenem C, se nachází tlačítko start, které spouští čtení dat z karty. To je ukončeno po naměření zadaného počtu otáček, nebo po clicku na vedlejší tlačítko stop. Poslední tlačítko save ukládá data do aplikace. Při zatrhnutí položky Run Time měření probíhá opakovaně dokud není přerušeno uživatelem pomocí tlačítka stop.
- D) Informace:** v části formuláře Rotation uniformity (viz obr. 3.21) označené písmenem D, se po naměření zobrazí základní informace o naměřených signálech. Jde o nosné složky ve spektrech IRC a tachosínálu, průměrné rychlosti otáčení a počtu naměřených otáček.
- E) Grafické znázornění:** v části formuláře Rotation uniformity (viz obr. 3.21) označené písmenem E, se po naměření dat vykreslují dle volby uživatele signály, které reprezentující vstupní data, mezivýsledky, výsledné signály i jejich frekvenční spektra.

Druhá záložka ve formuláři Rotation uniformity Card setting (viz obr. 3.23), se zabývá nastavením parametrů měřicí karty.



Obr. 3.23 Ukázka Formuláře nástroje Rotation uniformity – Záložka Card setting

- F) Chanel:** v části formuláře Rotation uniformity (viz obr. 3.23) označené písmenem E, je umožněno uživateli nastavit si vstupní kanály zařízení, tak aby odpovídali označení signálů v aplikaci.

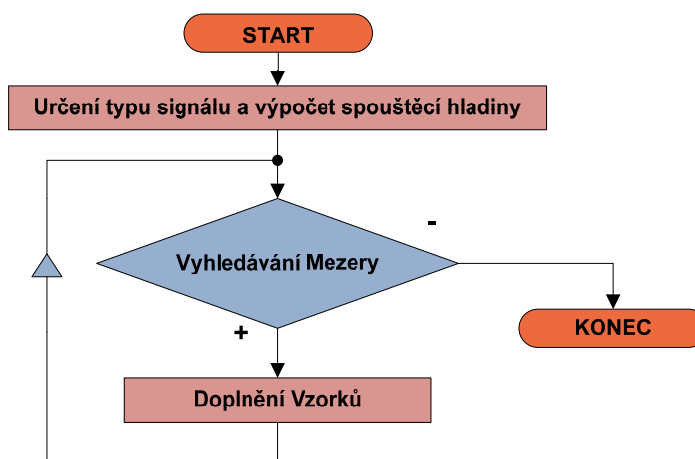
- G) Card setting:** v části formuláře Rotation uniformity (viz obr. 3.23) označené písmenem G, je možné nastavit jednotlivé parametry zvolené měřicí karty. Volbu této karty provedeme ve spodním comboboxu. O prvek výše uvádíme vzorkovací frekvenci a horním prvkem uživatel zvolí rozsah měření. Jeho správná volba je velice důležitá, protože ve chvíli měření ještě není známá maximální hodnota naměřeného signálu. Proto se pro výpočet spouštěcí hladiny, která slouží pro detekci otáček, využije tato hodnota
- H) Card Test:** v části formuláře Rotation uniformity (viz obr. 3.23) označené písmenem H, je možné pomocí clicku na tlačítko test provést kontrolní měření. To nám pomůže zjistit jaké vstupní signály jsou na jednotlivých vstupech karty, což velice usnadní nastavování parametrů.
- I) Grafické znázornění:** v části formuláře Rotation uniformity (viz obr. 3.23) označené písmenem I, dojde po ukončení kontrolního měření k vykreslení naměřených dat.



Obr. 3.24 Struktura algoritmu pro hodnocení rovnoměrnosti otáčení

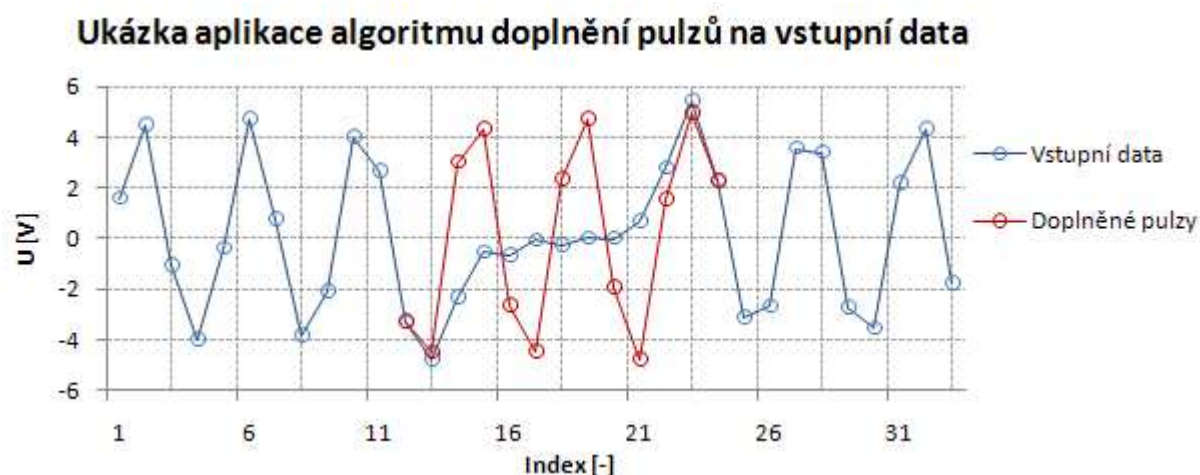
Popišme si nyní jak nástroj Rotation uniformity pracuje. Po nastavení parametru měřicí karty v záložce Card setting uživatel přejde na záložku Main. Zde spustí pomocí tlačítka start, po nastavení parametru nástroje Rotation uniformity, proces měření dat. Ten začíná výpočtem spouštěcí hladiny, která je vyčíslena jako procentuální hodnota (Položka Diagnostika spalovacích motorů na základě měření kolísání jejich úhlové rychlosti

Level v záložce Tachometer) z rozsahu měření (Záložka Card setting položka Range). Data jsou z měřicí karty načítány po menších dávkách. Přitom v datech odpovídajících tachosignálu jsou pravidelně kontrolovány detekce průchodu spouštěcí hladinou a dle nastavení tachometru inkrementovány otáčky. Po dosažení zadaného počtu otáček je měření ukončeno a přechází se na zpracování dat. Je možné, například u měření rovnoměrnosti otáčení spalovacích motorů, že v IRC signálu dojde k absenci několika pulzů. Proto je první část zpracování naměřených dat věnována jejich doplnění.



Obr. 3.25 Struktura algoritmu pro doplnění vzorků

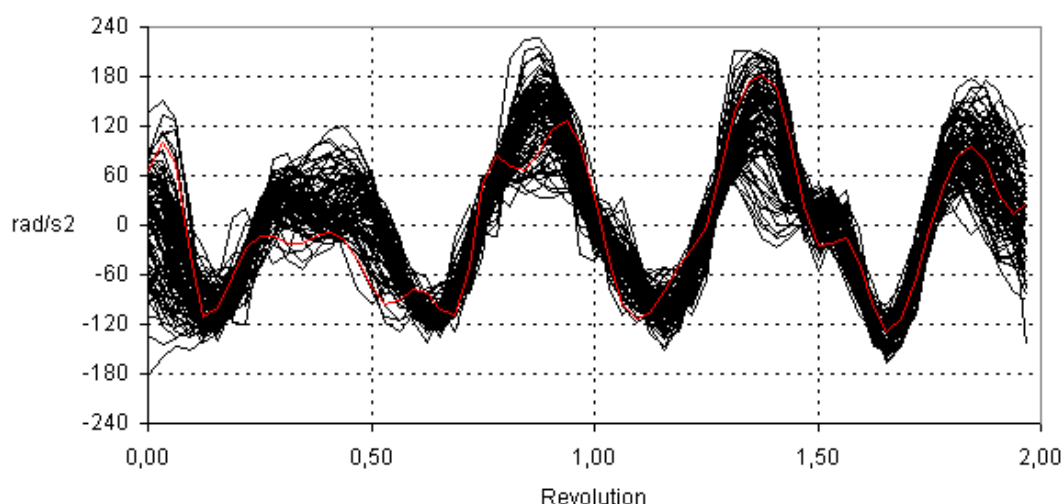
Algoritmus doplňování chybějících impulzů (viz obr. 3.25) vyhledává u každého dílčího impulzu parabolou interpolovanou buď maximum, nebo minimum a hledá časovou mezeru mezi těmito extrémy u sousedních impulzů. Tato mezera musí být větší, než $(\text{Počet chybějící pulzů} + 1) / 2$. Pokud je detekována mezera, je počítána počáteční fáze, ofset a amplituda nahrazující funkce kosinus. Ta bude mít v případě vynechání dvou impulzů tři periody (viz obr. 3.26). Algoritmus doplňování může pracovat správně, jen když jsou chybějící impulzy uvnitř záznamu.



Obr. 3.26 Ukázka aplikace algoritmu doplnění pulzů na vstupní data

Po doplnění chybějících impulzů dochází k úpravě vstupního signálu tak, že se za pomoci filtru pásmové propusti odstraní všechny složky spektra s výjimkou nosné složky a jejího postranního pásma. Je velmi vhodné, aby se šířka horního a dolního pásma rovnala polovině základní frekvence IRC signálu. Na této myšlence je založeno automatické nastavování filtru. Po odstranění nadbytečných složek spektra získáme harmonický fázově modulovaný signál. Následuje aplikace metody fázové demodulace, na jejímž konci získáváme odchylku úhlu od rovnoměrného otáčení (Phase detrended). Jednotlivé mezivýsledky této metody jsou uživateli zpřístupněny a to především z důvodu lepšího dohledání chyb v nastavení jednotlivých parametrů a podobně. Pro hodnocení okamžité rychlosti je nutné úhlovou výchylku derivovat podle času (Velocity). A pro určení zrychlení je potřeba dvojnásobné derivace (Acceleration). Derivace je v naší aplikaci řešena jako rozdíl aktuální hodnoty a hodnoty jí předešlé násobené převrácenou hodnotou délky časového intervalu mezi těmito vzorky.

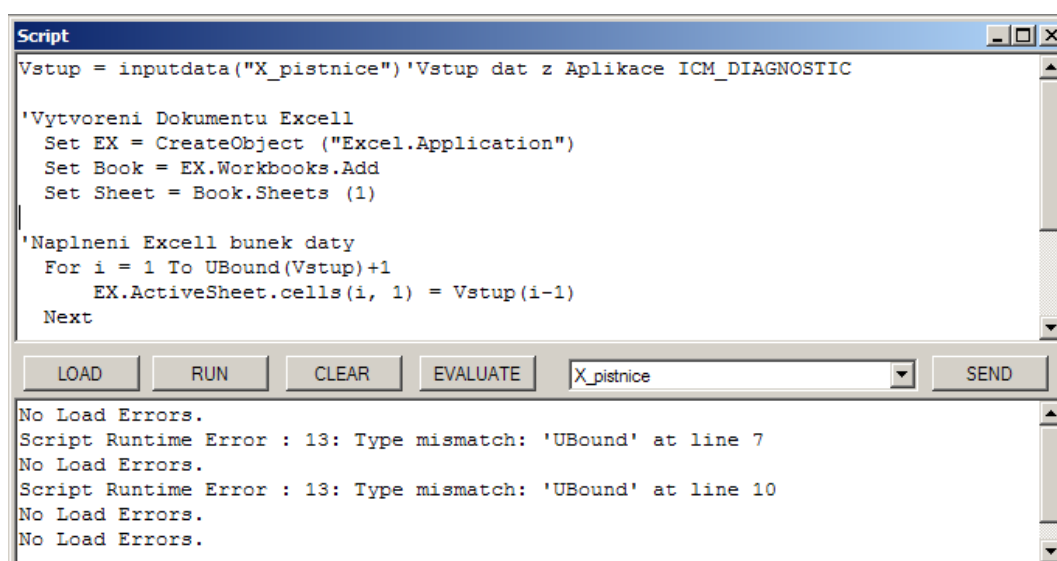
Způsob hodnocení rovnoměrnosti otáčení se mění případ od případu. Proto je možné vykreslit data v aplikaci jako celek, nebo po otáčkách. To je výhodné zvláště u diagnostiky spalovacích motorů. V tomto případě je také vhodné hodnotit rovnoměrnost otáčení pomocí úhlového zrychlení, protože jej lze spojit přímo s hnacím momentem stroje. Ukázka hodnocení rovnoměrnosti otáčení spalovacího motoru vyjádřené zrychlením a vykreslené po dvou otáčkách tak, aby byl pokryt celý pracovní cyklus je prezentována na obr. 3.27.



Obr. 3.27 Ukázka hodnocení rovnoměrnosti otáčení spalovacího motoru [TŮMA, J. a spol. 2002]

3.7 Script

Poslední prvek, který bude v této práci popsán, je Script (viz obr. 3.28). V tomto nástroji je uživatelem zadaný skript zpracován komponentou, kterou vytvořil Bc. Lukáš Plch. Ta je založena na Microsoft Script Control, která poskytuje skriptovací prostředí VBScript (Visual Basic – Scripting Edition). Tato komponenta je dále doplněna o další funkce pro zpracování vektorů a matic tak, aby usnadnila práci uživatelům, kteří nemají zkušenosti v oblasti programovacího jazyka Visual Basic.



Obr. 3.28 Ukázka nástroje Script

Pro pokročilejší uživatele pak tato komponenta umožňuje s výhodou používat veškeré prvky, které jazyk Visual Basic nabízí. Tím jsou myšleny například rozhodovací struktury jako If...Then, nebo Select case, dále cykly Do, while, For... To... Next a For Each... Next. Můžeme také využívat všechny operátory známé z VB a další prvky jako např. MsgBox, nebo InputBox. Dále je také možné vytvářet či přistupovat k objektům jiných aplikací, jako je Poznámkový blok, Word, Excel a mnoho dalších. Díky široké paletě doprogramovaných funkcí může tedy uživatel použít pro zpracování vektorů a matic následující operace:

- Sčítání, odečítání, násobení a dělení
- Použití matematických funkcí: odmocnina, exponenciála, logaritmus, absolutní hodnota, signum, zaokrouhlení, náhodné číslo
- Použití goniometrických funkcí: sinus, cosinus tangens, arkus tangens
- Výpočet DFT
- Návrh digitálních filtrů a jejich aplikaci na vstupní data
- Vykreslení dat v naší aplikaci a v aplikaci Excel
- Možnost nestále doplňovat paletu funkcí

Aby mohla být komponenta propojena s naší aplikací, bylo zapotřebí vytvořit společné rozhraní. Po vzájemné domluvě s autorem komponenty bylo za tímto účelem na její straně vytvořeno několik nových metod, událostí a pomocné pole, sloužící k předávání dat.

Předání skriptu komponentě:

```
' Preprocessing
code = "Sub Main()" & vbCrLf & txtboxInput.Text & vbCrLf & "End Sub"
code = Replace(code, "[", "GetMat(")
code = Replace(code, "]", "}")
' Metoda pro pridani kodu
AxSCRIPT1.AddCode(code)
```

Spuštění skriptu metodou Run:

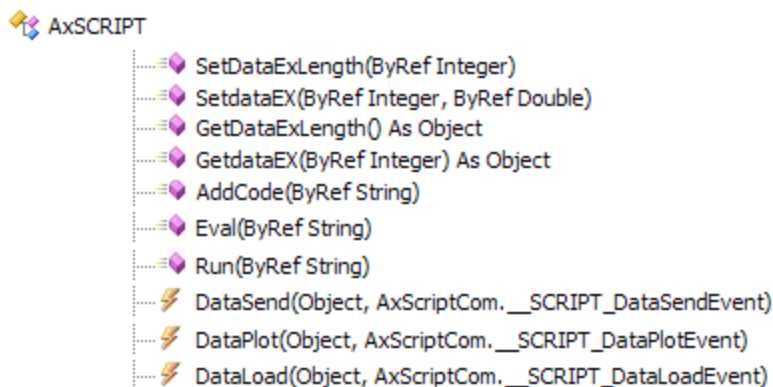
```
AxSCRIPT1.Run("Main")
```

Vyčíslení výrazu pomocí metody Eval:

```
AxSCRIPT1.Eval(txtboxInput.Text)
```

Textový výstup komponenty je vyřešen pomocí události Vystup:

```
Private Sub AxSCRIPT1_Vystup(ByVal sender As Object, _
                             ByVal e As AxScriptCom.__SCRIPT_VystupEvent) _
    Handles AxSCRIPT1.Vystup
    ' Parametr e.data ..... Vystup komponenty (Text)
    txtboxOutput.Text &= e.data & vbCrLf
End Sub
```



Obr. 3.29 Metody a události propojující komponentu s aplikací

Především metody a události jsou užity vždy, když chceme pracovat s nástrojem Script. Následující metody a události se týkají pouze použití určitých funkcí ve skriptu.

Načtení sinálu z aplikace do proměnné komponenty Script:

```
Vstup = inputdata("NazevSignalu")
```

Při zpracování výše uvedeného výrazu je v komponentě zavolána funkce `inputdata`. Ta v první řadě vyvolá událost `DataLoad`, jejímž parametrem je název signálu, který chceme nahrát do nástroje skript. Tato událost je zachycena v naší aplikaci. Zde proběhne průzkum, zda daný signál v aplikaci existuje. Pokud ano je připraveno pomocné

pole, do kterého je uložen signál. Následuje návrat do funkce `inputdata`, kde je signál z pomocného pole pomocí návratové hodnoty funkce předán proměnné `Vstup`. Takto nově vzniklá proměnná je typu `array` a je možné ji použít pod stejným názvem i v jiných částech skriptu, avšak až po uvedení výše zmiňovaného výrazu. Na podobném principu pracují i dále uvedené funkce.

Funkce pro uložení proměnné z komponenty Script do naší aplikace:

```
OutputData1 Vystup, "NazevNovehoSignalu"
```

```
OutputData2 Vystup, SampleRate, "NazevNovehoSignalu"
```

Parametr `Vystup` představuje proměnnou typu `array`, kterou chceme uložit do vlastní aplikace a `"NazevNovehoSignalu"` je jméno, pod kterým bude v aplikaci vystupovat. `SampleRate` udává jeho vzorkovací frekvenci. V případě `OutputData1` je nastavena na frekvenci 1024. Při použití předcházejících dvou funkcí je komponentou volána metoda `DataSend`.

Funkce pro vykreslení proměnné z komponenty Script:

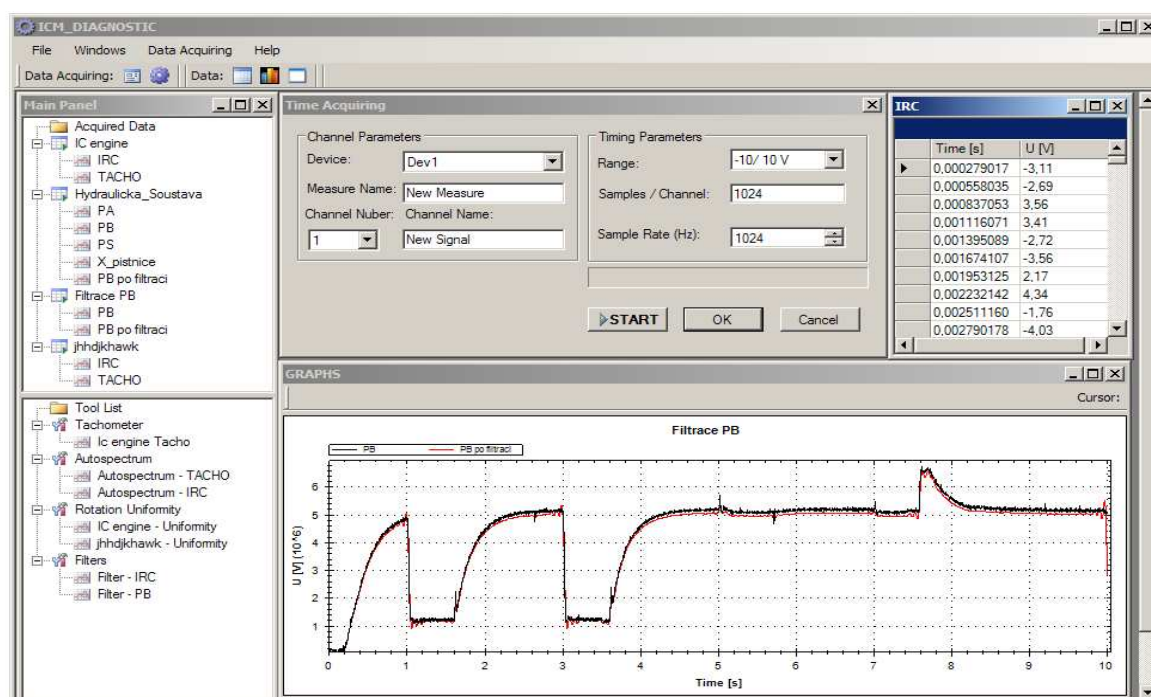
```
Plot1 Value
```

```
Plot2 Value, SampleRate
```

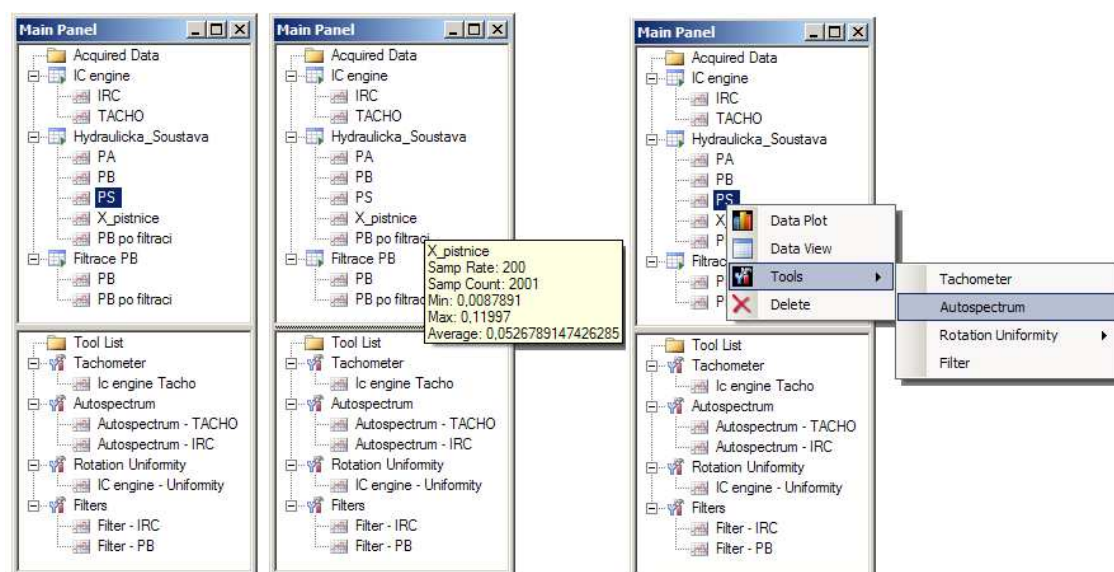
Parametr `Value` představuje proměnnou typu `array`, kterou máme v úmyslu vykreslit a `SampleRate` udává jako stejně jako v předchozím případě vzorkovací frekvenci. Ta je u `Plot1` nastavena na hodnotu 1. Při použití předcházejících dvou funkcí je komponentou volána metoda `DataPlot`.

3.8 Uživatelské rozhraní

Vytvořené nástroje bylo třeba sjednotit v jedné aplikaci. Za tímto účelem tedy bylo vytvořeno uživatelské prostředí, které poskytuje uživateli uložení a organizaci naměřených dat, možnost aplikovat na tyto data již dříve popsání nástroje a jejich organizaci. Přehledné zobrazení dat v tabulce nebo grafu, import a export dat do jiných aplikací pomocí schránky a také možnost ukládat vytvořené projekty na pevný disk. Představme si tedy jednotlivé prvky tohoto prostředí.

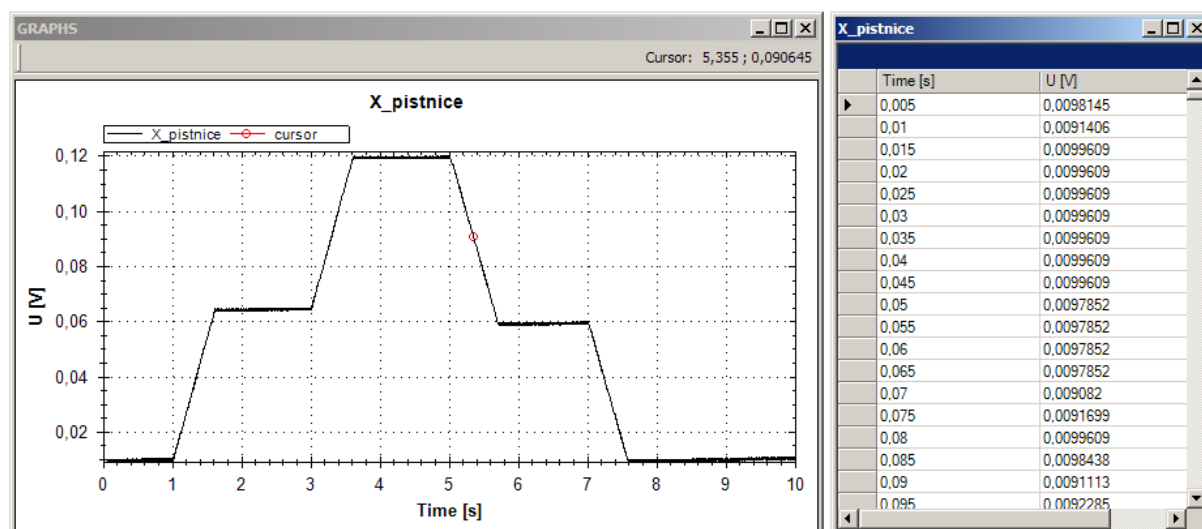


Obr. 3.30 Ukázka vytvořené aplikace zajišťující sběr a zpracování dat



Obr. 3.31 Ukázka organizéru Nástrojů a naměřených dat MainPanel

Na obr. 3.31 je znázorněn formulář pro organizaci nástrojů a naměřených dat MainPanel. Prostřední obrázek zde prezentuje zobrazení základních vlastností signálu pomocí tooltipu, který je zobrazen při zaměření na položku ve stromové struktuře signálu. Na posledním obrázku je prezentována nabídka, pomocí které lze signál přiřadit zvolenému nástroji, smazat či vykreslit.



Obr. 3.32 Ukázka zobrazení dat v grafu a pomocí datové tabulky

Jelikož je pro program zabývající se zpracováním signálů téměř nezbytné grafické vyjádření dat, bylo nutností poohlédnout se po komponentě zajišťující tuto potřebu. Většinu grafických komponent nabízel komerční sektor. V rámci minimalizace nákladů na vytvoření celé aplikace bylo hledání soustředěno na sektor nekomerční. Zde byla objevena komponenta pro vykreslování grafů pod názvem ZedGraph. Pomocí této komponenty je možné vykreslovat všechny běžné typy 2D grafů, např. sloupcový, spojnicový, výsečový a mnoho dalších. Za velkou výhodu této komponenty je považována rychlá možnost zvětšení vybraných oblastí grafu v uživatelském prostředí a jednoduchá a intuitivní aplikace do vlastní aplikace. Jednoduchost vytvoření grafu ve vlastní aplikaci demonstruje následující ukázka:

```
' Import knihovny
Imports ZedGraph
.
.
' Ukázková data
Dim DataX() As Double = {1, 2, 3, 4, 5}
Dim DataY() As Double = {3, 5, 10, 8, 4}
' Nastavení popisu os
Graf1.Title.Text = "Vzorový příklad"
Graf1.XAxis.Title.Text = "Popis Osy X"
Graf1.YAxis.Title.Text = "Popis Osy Y"
' Zobrazení mřížky
Graf1.GraphPane.XAxis.MajorGrid.IsVisible = True
Graf1.GraphPane.YAxis.MajorGrid.IsVisible = True
```

```
' Přidání spojnicového grafu
```

```
Graf1.GraphPane.AddCurve("Spojnicový graf Ukázka", DataX, DataY, _  
                          Color.Red, SymbolType.Circle)
```

```
' Přidání sloupcového grafu
```

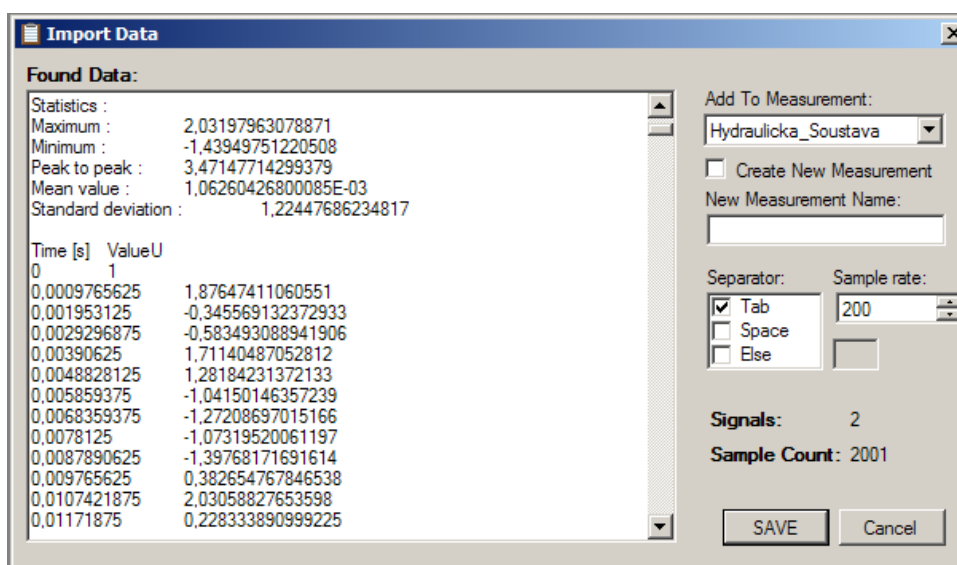
```
Graf1.GraphPane.AddBar("Sloupcový graf Ukázka", DataX, DataY, _  
                       Color.Blue)
```

```
' Přepočet os
```

```
Graf1.AxisChange()
```

```
.
```

Z datové tabulky je poté možné pomocí schránky exportovat data do jiných aplikací.

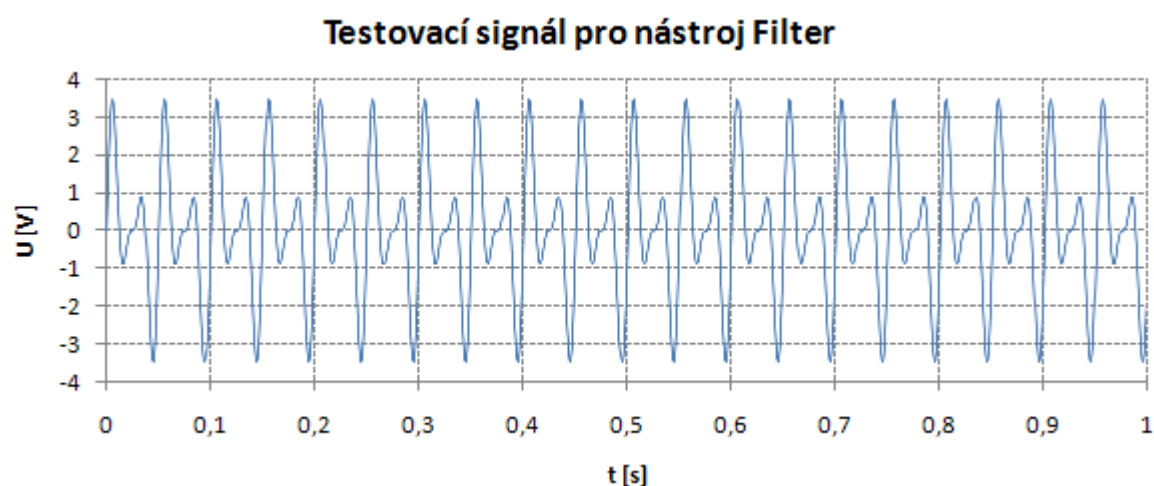


Obr. 3.33 Formulář doprovázející import dat ze schránky

Z aplikace je možné data přes schránku nejen exportovat, ale i importovat. Import dat ze schránky doprovází formulář Import data prezentovaný na obr. 3.33. Je zde možné zvolit oddělovač dat, vzorkovací frekvenci, kterou byla data pořízena a měření, ke kterému chceme data připojit, nebo jméno nově vzniklého měření. Algoritmus importu je navržen tak, aby vynechal hlavičku před a volné řádky na koci datových sloupců.

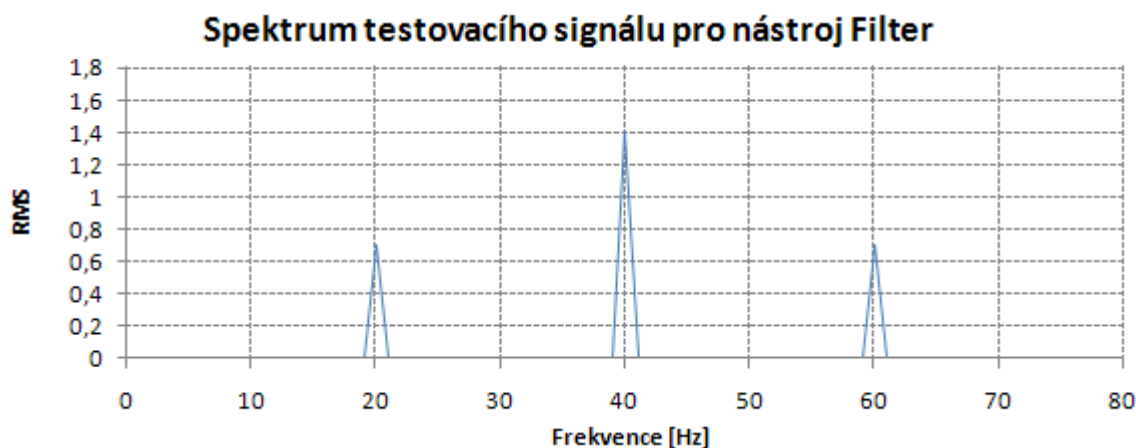
4 Demonstrace funkčních možností

Poslední díl práce byl věnován ověření funkceschopnosti diagnostického systému jako celku. Předmětem této kapitoly je tedy popis postupu ověřování vybraných nástrojů. Ověřování začne u nástroje Filter. Ten bude ověřen společně s nástrojem Autospectrum, jelikož je zde zapotřebí zobrazit jak testovací signál, tak i výsledek ve frekvenční oblasti.



Obr. 4.1 Testovací signál pro nástroj Filter

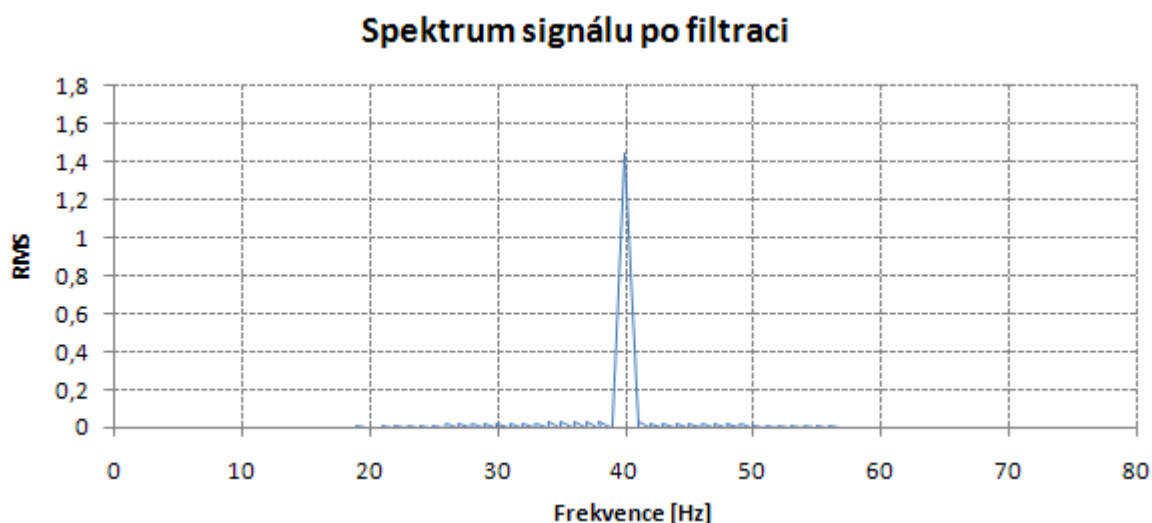
Jako testovací signál byl zvolen signál, který se skládá ze tří sinusových signálů o frekvencích 20, 40 a 60 hertzů. Jednotlivé složky je možné vidět na obr. 4.2. Jeho časový průběh je prezentován na obr. 4.1.



Obr. 4.2 Spektrum testovacího signálu pro nástroj Filter

Na tento signál byl následně aplikován filtr typu pásmová propust. Jeho střed byl nastaven na frekvenci 40 hertzů a šířka propustného pásma na 20 hertzů tak, aby ve výsledném signálu zůstala po filtraci pouze složka o frekvenci 40 hertzů. Důkazem toho, že zkouška nástroje filtr proběhla v pořádku, je obr. 4.3 znázorňující spektrum výstupního signálu zvoleného filtru. To obsahuje pouze složku o frekvenci 40 hertzů, která ležela

v propustném pásmu. Složky o frekvenci 20 a 60 hertzů, ležící v nepropustném pásmu byly filtrem odstraněny. Funkčnost se potvrdila též u nástroje Autospectrum.



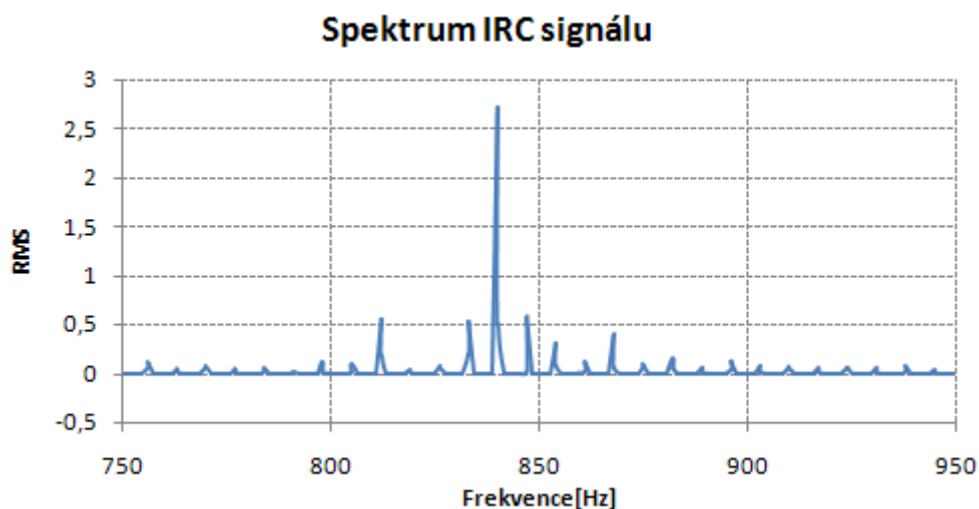
Obr. 4.3 Spektrum signálu po filtraci

Dalším nástrojem vytvořeným v této aplikaci je nástroj Tachometer. Popis testování tohoto nástroje je uveden již v předchozí kapitole. Přejdeme tedy rovnou k nástroji Rotation uniformity. Funkčnost tohoto nástroje se potvrdila již při vývoji, kdy byly na tento nástroj přiváděny vygenerované fázově modulované harmonické signály. V této kapitole se zaměříme testováním komponenty na reálných datech. Měření se zabývá rovnoměrností otáčení spalovacího motoru. Zde byl využit jako zdroj tachosignálu výstup snímače, který je umístěn na vačkovém hřídeli a IRC signál byl získán na výstupu snímače umístěném na setrvačnicku motoru. Ukázka IRC signálu je prezentována na obr. 4.4.

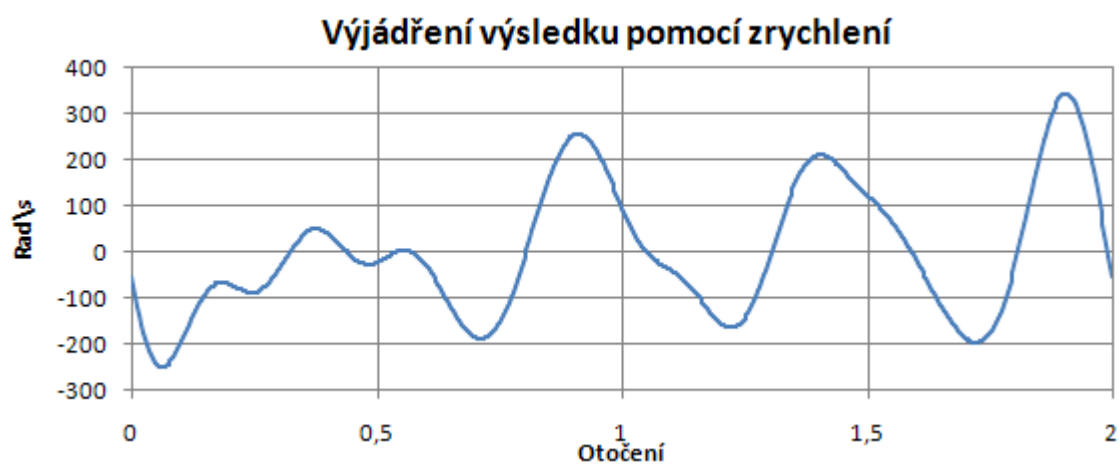


Obr. 4.4 Ukázka výstupního signálu senzoru na setrvačnicku

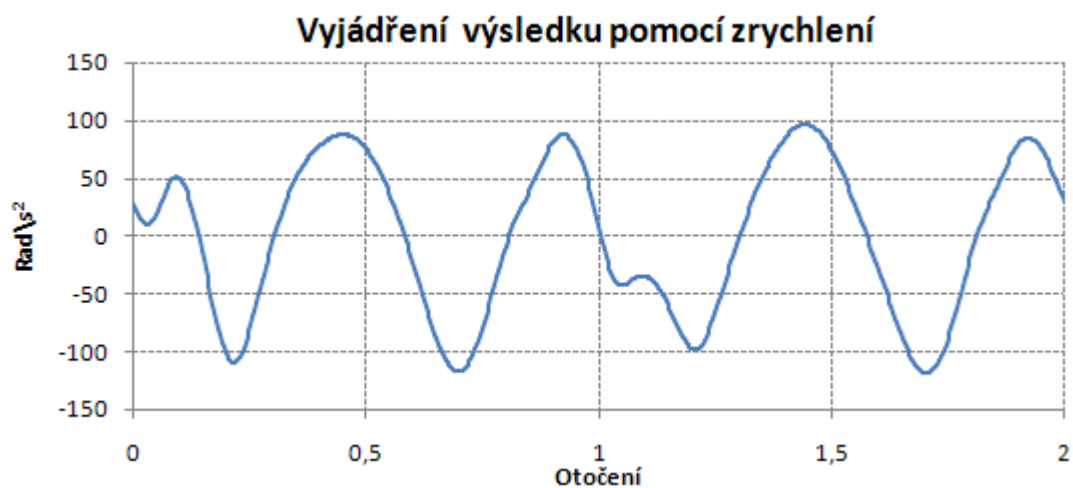
Následující grafy již prezentují výsledky výstupu nástroje Rotation uniformity. První viz obr. 4.5 prezentuje výpočet spektra IRC signálu. Druhý graf viz obr. 4.6 nabízí ukázkou nerovnoměrného pohybu a poslední viz obr. 4.7 nabízí ukázkou pohybu rovnoměrného.



Obr. 4.5 Ukázka výpočtu spektra signálu



Obr. 4.6 Ukázka Výsledku hodnocení rovnoměrnosti otáčení spalovacího motoru



Obr. 4.7 Ukázka Výsledku hodnocení rovnoměrnosti otáčení spalovacího motoru

5 Závěr

Hlavním cílem této práce bylo vytvořit diagnostický systém pro hodnocení rovnoměrnosti otáčení. Zadanou problematiku jsem řešil následovně.

Zaměřil jsem se na analýzu měření úhlové rychlosti pomocí IRC snímačů a Dopplerovského laseru. Seznámil jsem se a popsal základní vlastnosti těchto snímačů, fyzikální princip či způsob detekce směru otáčení. Dále jsem nastudoval a popsal základní vlastnosti a parametry měřicí karty NI USB-6009. Zahrnul jsem zde i popis vstupních a výstupních kanálů. Další studium jsem věnoval metodě fázové demodulace analogového signálu, jejíž popis je součástí této práce. Následně jsem navrhl měřicí řetězec i s jeho hardwarovým vybavením, jehož hlavními prvky jsou IRC snímač a měřicí karta NI USB-6009.

Po návrhu technických prostředků následovala tvorba softwarového zajištění měřicí úlohy. Před vytvořením vlastní aplikace jsem se seznámil s vývojovými prostředími vhodnými pro tvorbu měřících aplikací, podporující zvolený typ měřicí karty. Zvoleno bylo prostředí Microsoft Visual Studio 2008, programovací jazyk Visual Basic a platforma Microsoft .NET Framework.

Ve výsledném softwaru jsem vytvořil propojení s měřicí kartou NI USB-6009 pomocí knihovny NIDAQmx.dll. Pro zpracování naměřených dat jsem vytvořil tyto virtuální softwarové nástroje: První nástroj pod názvem Tachometr, který slouží pro získání velikosti nominální rychlosti otáčení. Tento nástroj se používá pro zpracování pulzního signálu. Ten je možné získat na výstupu tachosondy, IRC snímač, nebo v případě spalovacího motoru snímačem umístěným na vačkovém hřídeli.

Dalším nástrojem je Autospectrum, který slouží k zobrazení frekvenčního spektra signálů. Pro výpočet diskrétní Fourierovy transformace jsem použil komponentu založenou na algoritmu the Fastest Fourier Transform in the West. K této komponentě jsem doprogramoval časová okna, přepočít měřítka osy x na frekvenci v hertzech a pro amplitudu jsem vytvořil tyto tři měřítka: RMS (Efektivní hodnota), PWR (Celkový výkon signálu) a PSD (Výkonová spektrální hustota).

Dalším nástrojem je Filter. Ten využívá digitálních filtrů pro odstranění uživatelem vybraných frekvenčních složek vstupního signálu. V tomto nástroji jsem vytvořil algoritmy pro návrh koeficientů filtru, algoritmus aplikace filtru na vstupní signál a zobrazení frekvenční odezvy filtru, která usnadňuje jeho návrh.

Dalším nástrojem, který jsem vytvořil, je Rotation uniformity sloužící pro hodnocení rovnoměrnosti otáčení. Ten je založen na metodě fázové demodulace a využívá předešlých nástrojů tak, aby byly soustředěny na jednom místě všechny potřebné informace k vyhodnocení daného problému. Aplikace obsahuje i nástroj Skript. Ten umožňuje uživateli tvorbou vlastních funkcí a algoritmů v programovém jazyce VBScript. Je zde také možné využít mnoho předprogramovaných funkcí pro zpracování vektorů a matic. Nástroj tedy nabízí uživateli mnoho dalších možností při zpracování signálů. Má práce zde spočívala ve vytvoření uživatelského rozhraní a propojení komponenty s aplikací.

Vykreslování grafů jsem zajistil aplikací komponenty pod názvem ZedGraph. Data je možné zobrazit také v datové tabulce. Pomocí schránky lze tyto data exportovat i do jiných aplikací. Pomocí schránky je umožněn i import dat. Pro tyto účely jsem vytvořil doprovodný formulář. Nastavení a data představují projekty, které je možné ukládat na pevný disk.

Funkčnost jednotlivých prvků aplikace jsem postupně prověřoval pomocí kontrolních měření jak na reálných, tak i na vygenerovaných signálech. Při těchto měřeních se potvrdila funkčnost vytvořených nástrojů. Následující práce by měla být zaměřena na vytváření nových virtuálních nástrojů, což rozšíří okruh použití vytvořené aplikace.

6 Použitá literatura

FARANA, R., SMUTNÝ, L., VÍTEČEK, A. & VÍTEČKOVÁ, M. 2004. *Zpracování závěrečných prací z oblasti automatizace a informatiky*. Ostrava: VŠB, 2004, 114 s. ISBN 80-248-0557-X

GURTNER, M. & KOCICH, P., 2005. *Visual basic .NET Hotová řešení*. Brno: Computer Press a.s., 2005, 312 s. ISBN 80-251-0367-6

HALLIWELL N. A., ROTHBERG S. J. & MILES T. J. 2008. *On the Working Principle of Torsional Vibration MeterType 2523*, [on line]. [cit. 2008-05-10]. Available from World Wide Web: < <http://www.bksv.com/doc/bo0426.pdf>>

HALVORSON, M., 2008. *Visual basic 2008 krok za krokem*. Brno: Computer Press a.s., 2008, 440 s. ISBN 80-251-2221-1

HEIDENHAIN. 2008. *Bidirectional Interface for Position Encoders*, [on line] [cit. 2008-02-6] Available from World Wide Web: <http://www.heidenhain.com/index.php?WCMSGGroup_2243_177=846&WCMSGGroup_2238_177=2243&WCMSGGroup_846_177=4802>

KAČMÁŘ, D. & FARANA, R. 1998. *Vybrané algoritmy zpracování informací*. Ostrava: VŠB, 1998, 114 s. ISBN 80-7078-398-2

KLOS O. 2008. *Nové inkrementální rotační snímače ifm electronic*, [on line] [cit. 2008-05-8]. Available from World Wide Web: < <http://www.automatizace.cz/article.php?a=990>>

KMIT, P. 2008. *Měření prostřednictvím USB*, [on line]. [cit. 2008-05-4]. Available from World Wide Web: < <http://www.automatizace.cz/article.php?a=881>>

KREIDL M. & ŠMÍD R. 2006. *Technická diagnostika - senzory, metody, analýza signálu*, Praha: BEN, 30.11.2006, s. 408. ISBN 80-7300-158-6

NATIONAL INSTRUMENTS. 2008. *Online katalog*, [on line]. [cit. 2008-05-1] Available from World Wide Web: < <http://sine.ni.com/nips/cds/view/p/lang/en/nid/201987>>

POLYTEC. 2008. *Made Possible by Laser Vibrometry*, [on line]. [cit. 2008-04-4]. Available from World Wide Web: <www.polytec.com/usa/_files/LM_AN_INFO_0103_E_Vibrometry_Basics.pdf>

SMUTNÝ, L., TŮMA, J. & FARANA, R. 2006. *Calibration of Sensors for Angular Vibration Measurements*. In XVIII IMEKO WORLD CONGRESS „Metrology for a Sustainable Development“. Rio de Janeiro, Brazil: IMEKO, 17. – 22. 9. 2006, 4 p.

STROUSTRUP, B., 1997. *C++ Programovací jazyk*. Praha: BEN, 1997, 686 s. ISBN 80-901507-2-1

TŮMA, J. 2008. *Autospectrum*, [on line]. [cit. 2008-12-5] Available from World Wide Web: < <http://fs1.vsb.cz/~tum52/SignalAnalyser/Prezentace/Autospectrum.pdf>>

TŮMA, J. 2009 *Digital filter*, [on line]. [cit. 2009-04-1] Available from World Wide Web: < http://fs1.vsb.cz/~tum52/SignalAnalyser/Prezentace/Filtry_EN.pdf >

TŮMA, J. 2001. *Měření okamžitých otáček úhlovým snímačem otočení*. In Technická diagnostika strojů a výrobních zařízení „DIAGO 2002“, Ostrava: VŠB-TU Ostrava, 30. – 31. leden 2001. p. 458-464. ISBN 80-248-0045-4.

TŮMA, J. 2002. *Phase demodulation in angular vibration measurements*. In Proceedings of 3rd International Carpathian Control Conference. Ostrava: VŠB-TU Ostrava, 27. - 30. 5. 2002, s. 99-104. ISBN 80-248-0089-6.

TŮMA, J., KOČÍ, P., ŠKUTA, J. & JURÁK, M. 2002. *Analysis of a car burst shaking while the car engine is running at idle*. In Proceedings of 3rd International Carpathian Control Conference. Ostrava: VŠB-TU Ostrava, 27. - 30. 5. 2002, s. 105-110. ISBN 80-248-0089-6.